Towards Robust Visual Speech Recognition

Automatic Systems for Lip Reading of Dutch

Towards Robust Visual Speech Recognition

Automatic Systems for Lip Reading of Dutch

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op dinsdag 2 november 2010 om 12.30 uur
door

Alin Gavril CHIŢU

Master of Science in Computer Science, Universitatea Bucureşti,
Master of Science (ir) in Applied Mathematics, Technische Universiteit Delft
geboren te Buşteni, Roemenië.

Dit proefschrift is goedgekeurd door de promotoren:

Prof. dr. C.M. Jonker
Prof. dr. drs. L.J.M. Rothkrantz

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. C. M. Jonker,	Delft University of Technology, promotor
Prof. dr. drs. L. J. M. Rothkrantz,	Netherlands Defence Academy, promotor
Prof. dr. ir. F. W. Jansen	Delft University of Technology
Prof. dr. I. Heynderickx,	Delft University of Technology
Prof. dr. I. Văduva,	University of Bucharest
Prof. dr. Ing. Elmar Nöth,	University of Erlangen-Nürnberg
Dr. K. C. H. M. Nieuwenhuis,	DECIS Lab & TRT-NL

The work was carried out in the Man-Machine Interaction Group in the Mediamatics Department at Delft University of Technology and was part of the Interactive Collaborative Information Systems (ICIS) project financed by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

ISBN: 978-90-8570-697-7

Printed in The Netherlands by Wöhrmann Print Service.

To my dearest wife Dana
and our sweetest daughters Ana and Mina,

Contents

Chapter 1

Introduction

1.1 Lip Reading by Humans

Lip reading was thought for many years to be specific to hearing impaired persons. Therefore, it was considered that lip reading is one possible solution to an abnormal situation. Even the name of the domain suggests that lip reading was considered to be a rather artificial way of communication because it associates lip reading with the written language which is a relatively new cultural phenomenon and is not an evolutionary inherent ability. Extensive lip reading research was primarily done in order to improve the teaching methodology for hearing impaired persons to increase their chances for integration in the society. Later on, the research done in human perception and more exactly in speech perception proved that lip reading is actively employed in different degrees by all humans irrespective to their hearing capacity. The most well know study in this respect was performed by Harry McGurk and John MacDonald in 1976. In their experiment the two researchers were trying to understand the perception of speech by children. Their finding, now called the McGurk effect, published in Nature [Mcg76], was that if a person is presented a video sequence with a certain utterance (i.e. in their experiments utterance 'ga'), but in the same time the acoustics present a different utterance (i.e. in their experiments the sound 'ba'), in a large majority of cases the person will perceive a third utterance (i.e. in this case 'da'). Subsequent experiments showed that this is true as well for longer utterances and that is not a particularity of the visual and aural senses but also true for other perception functions. Therefore, lip reading is part of our multi-sensory speech perception process and could be better named visual speech recognition. Being an evolutionary acquired capacity, same as speech perception, some scientists consider the lip reading's neural mechanism the one that enables humans to achieve high literacy skills with relative easiness [Att06].

Another source of confusion is the “lip” word, because it implies that the lips are the only part of the speaker face that transmit information about what is being said. The teeth, the tongue and the cavity were shown to be of great importance for lip reading by humans ([Wil98a]). Also other face elements were shown to be important during face to face communication; however, their exact influence is not completely elucidated. During experiments in which a gaze tracker was used to track the speaker's areas of attention during communication it was found that the human lip readers focus on four major areas: the mouth, the eyes and the centre of the face depending on the task and the noise level ([Buc07]). In normal situations the listener scans the mouth and the other areas relatively equal periods of times. However, when the background noise increases, the centre of the face becomes the central point of attention. Most probably the peripheral vision becomes extremely active in these situations. When the task was set to the inference of the emotional load of the interlocutor, the listener's gaze started to be shifted towards the eyes since they convey more emotional related information. It is well accepted that the human lip readers make great use of the context in which the interaction takes place. This can be one of the reasons the human listener scans the entire face during the interaction. In [Hil09] the authors found that when a human lip reader was presented with appearance information, compared with only mouth shapes, his performance increased considerably from 42.9% to 71.6%.

We should realise that during face to face interaction a human engages in a complex process which involves various channels of information corresponding to our senses. In this way the speaker builds up the context using both verbal and non-verbal cues such as body gesture, facial expressions, prosody, and other physiological manifestations. Other information about the settings in which the communication takes place is used as well as the knowledge accumulated in time through experience. A human is a multi-modality, multi-sensory, multi-media fusing machine.

1.2 Automatic Lip Reading

Automatic lip reading has emerged as a research domain relatively late in the 1980s. Having a slow start due to the lack of computational power at the time, the interest into this domain has increased considerably starting with the late 1990s. In the beginning it was seen as a natural way to increase the more mature speech recognition systems, later on however, the stand alone value was also considered. In the literature automatic lip reading is referred to in many ways, with or without the word "automatic", some of which are: lipreading, speech-reading, lip-reading, speech-reading, optically based speech recognition, visually based speech recognition, audio-visual speech recognition, etc. In this dissertation "lip reading" is in general used instead of "automatic lip reading".

With the increase in the computational power and the generalized access to cheaper and powerful information systems (in the less developed parts of the world the mobile phones play an important role; mobile cellular subscriptions worldwide is approaching 5 billion at the time of this writing), there is an increased need, but also increased affordability, to improve the interface between the human users and their artificial assistants. The goal is to make the interaction easier for a larger range of users, therefore coping with the personal needs of more categories of users, and closer to the natural ways of communication of humans. Therefore, the next generation of user interfaces should permit communication through speech and lip reading, through body gesture, but also other means. Another, very active connected research domain is on emotional state recognition which would enable the system adaptation to the current state of the user. With the emergence of the "Ubiquitous computing" paradigm human-computer interaction goes to a different level, where the information processing units are thoroughly embedded into everyday objects. The next generation of user interfaces will need to sense and understand the environment and its users with the use of natural ways of communication. The artificial systems of tomorrow will be omnipresent but in the same time less visible to the user, and therefore, it will be necessary that the interaction be possible with a user that is unaware of their presence.

1.3 Societal Relevance - Applications

1.3.1 Crisis Management - The ICIS Project

After the events on September 11, 2001, the need for developing more sophisticated techniques for improving the response and management of crises situations has become evident. The present research was done as part of the Interactive Collaborative Information Systems (ICIS)[D-C04] project supported by the Dutch Ministry of Economic Affairs. The goal of ICIS was to develop better techniques for making complex information systems more intelligent and supportive in critical decision making situations. Although the research results generated in the project are generally applicable, ICIS was focused on two key applications domains: traffic management and crisis response. In the case of the crisis management application, one key element that could be sustained through intelligent systems was the communication among the actors in the field and the communication between the decision units and the acting units [Fit07]. During a crisis event the performance of the acoustically based speech recognition drastically decreases due to the large background noise. Therefore, lip reading is necessary in order to make this important means of communication usable in these situations[Fit10].

1.3.2 Lip Reading Applications

As we mentioned before, the first application of automatic lip reading was to improve automatic speech recognition. This means fusing acoustic information with visual information in order to stabilise the performance in case of increased background noise. Fusing multi-modal data is a research topic in itself, and is beyond the scope of this thesis.

Later on, lip reading was also seen as stand alone application. As stand alone, automatic lip reading's first choice application is improving the interaction of the hearing impaired persons. A phone enhanced with lip reading and talking face capabilities would introduce this communication technology, which is so common for the rest of us, to speaking impaired persons. For instance the SynFace [syn01] project aimed to develop a real-time system able to deliver phonetically informative lip reading cues derived from parameters extracted from the acoustic speech signal to assist people during phone conversations. Recent mobile phone models add lip reading technology in order to make possible the use of the phone in places where the etiquette does not permit load speech (e.g. in a theatre). This is a part of the so called silent speech interface endeavour [Sch10].

Another application which recently received great interest is the recovery of speech in the deteriorated or completely mute video footage. The lip reading technology was applied on archived home videos of Adolf Hitler. Filmed by Eva Braun during the war, the films were recorded with the technology of their time and provided no sound. The recovery of the speech in these films can be important from a historical point of view, since it could provide new insights on the dictator's life.

In the previous section we discussed lip reading technology in the context of crisis management. Similarly, a newer idea is to use lip reading for security applications. Lip reading can empower surveillance systems with the means of recovering speech information from a distance even when there are no audio devices.

In general, lip reading will be an integrant part of the next generation user interfaces.

1.4 Problem Definition and Objective

Even though more than 20 years passed since its birth, the lip reading research community is still searching for its giant leap. Starting as a complementary process for speech recognition, it largely remained, until now, in its shadow. With the increase in the available computational power and the appearance of new applications for lip reading this situation is going to rapidly change.

One of the main problems with lip reading is the fact that there is still no consensus in the scientific community on the parametrization that yields the highest performance. To this moment it is still not well understood how people are lip reading and, therefore, where the information on what is being said can be found.

The fact that there is no underling model that defines the basic structures on which recognition systems are to be built is an equally important problem. While in the acoustic domain the theory of phones and phonemes is mature and has a profound structure and understanding, in the visual domain the definition of the basic elements, the visemes, is relatively vague. A viseme is a set of phonemes which are easily confused by a human listener. In some sense we have a pseudo-definition since the viseme sets are not uniquely defined and are interpretable at best. This approach not only makes it difficult to interpret and compare the results of different experiments but also introduces a theoretical upper limit to the performance of the lip reader which is always a fraction of the performance of the speech reader.

Other problems, which are still proving to be restrictive in spite of the advances in processing power, bandwidth and storage space, are the data acquisition, data annotation and data parametrization. The amount of man hours is still too high for these problems to be easily surmounted. This is because not all the processes involved in data acquisition can be automatized. Therefore, the data corpora for lip reading are still not large enough to train continuous speech lip readers. Also, with respect to data parametrization the available computational power does not always suffice in order to obtain real time systems.

As in other computer vision related domains, there are big problems with the processing of the data. These problems come from the instability of the pattern recognition algorithms which are still not fully developed in order to deal with the visual artefacts induced by illumination variation and occlusion. In real life situations the problems of robust detection and tracking of objects are not yet solved. These problems are beyond the scope of this thesis but they greatly influence the performance of automatic lip reading and, therefore, need to be acknowledged.

Dealing with all these problems lip reading might still remain for some time in the shadow of acoustic speech recognition. However, we should remember that the stated goal of lip reading at its beginnings was exactly this, to act as an enhancer for speech recognition. Therefore, it is not needed to have perfect lip reading if we can achieve the goal of improving the speech recognition process in general. In this sense lip reading is only one of many other directions that can bring valuable insights for speech recognition (e.g. emotion recognition, gesture and body posture recognition, topic recognition and in general terms any other method that can provide context awareness to the system). Fusing information coming from multiple sources is another research topic in itself, and is beyond the topic of this thesis. Lip reading is one more important step towards a multi-modal speech recognizer and, extrapolating, towards a multi sensorial interaction with our artificial partners. This is probably our highest goal after all, to achieve the same level of communication as with human interlocutors. This would enable entirely new applications.

However, there are situations when the importance of the lip reader increases considerably. Namely, in noisy environments where the acoustic environment becomes unusable, the lip reader may entirely replace the speech recognizer. Therefore, it is necessary to investigate the limits of lip reading and also the possibilities to boost its performance.

The endeavour of systems developers to ease the communication with the human users, but also to accommodate a larger group of human users is consistent with our desire to continuously improve the quality of life. We assist to the improvement of the quality of life using a continuously broadening set of tools.

The main research question that this thesis addresses is:

What are the possibilities to boost the performance of a lip reader such that a robust recognition system is obtained?

In order to answer this question we investigated the limitations of the current approaches and the state of the art and contributed with various solutions that improved the process of building a lip reader (e.g. we standardized the data acquisition process) and boosted the lip reading performance (e.g. we investigated various data parameterisations techniques and we analyzed the lip reading performances under various settings). More detailed research questions that this thesis addresses are:

1. Are we able to build a lip reader for the Dutch language? This lip reader should be able to achieve good performance (i.e. comparable with or higher than the performance obtained by other researchers for other languages) on tasks of various complexity.

2. For speech recognition the most successful approach, to date, is the Bayesian probabilistic approach Hidden Markov Models. Is this approach suitable for lip reading as well?

3. What are the problems with the existing data corpora for lip reading?

4. Can we write a set of guidelines for building a data corpus for lip reading?

5. Can we create a corpus for the Dutch language large enough to support our endeavour? Due to the circumstances, this is a great step towards our goal to build a robust lip reader in Dutch. The corpus should be a good basis for future research in lip reading and connected domains.

6. What is the influence of the data parametrization method on the performance of the lip reader? We want to test different methods and compare their performances on similar recognition tasks. What are the strengths of the various feature extraction methods?

7. Should we concentrate on the contour of the mouth, or is better to have a broader appearance based approach?

8. How important is the motion? Do we need to compute the motion flow on the speaker's face or it is sufficient to generate a static model of the speaker's face and only compute the derivatives based on this model?

9. What is the influence of the speaking style on the performance of the lip reader? The speech rate was never considered as an issue in lip reading even though common sense tells us that it has great influence on the way people speak.

10. Is there any correlation between the speaking style and the choice of the data parametrization used, namely is one parametrization more suitable for some speaking style than the others?

11. Do we need to use a higher recording rate on the visual side? For a long time, the video sampling rate (e.g. 25Hz) was seen as a performance impediment because it did not match the audio sampling rate (i.e. usually 100Hz). However, there was no rigorous study of the influence of the video sampling rate on the performance of the automatic lip reading.

12. What is the influence of the size of the corpus on the performance of the resulting recognition system?

13. What is the best definition of the visemes? Visemes are the phonemes counterparts in the lip reading domain. Usually, visemes are defined by elicitation. However, this is beyond the scope of this thesis. What we want is to have a thorough literature survey on the papers that cover this topic for the Dutch language.

Table 1.1 summarizes the main trajectory of our approach in order to answer the research questions tackled in this thesis. A more detailed description of the methodology we used, and justifications for the actions we have taken are found in Chapter 2. In general, we approached the problems in a problem-experiments-analysis-fine-tuning fashion. Each time we started by designing suitable experiments which we exhaustively fine-tuned until the best results are obtained for the given task.

Table 1.1: *The main trajectory of the thesis.*

	Action	Result
1	Literature survey	State of the Art of lip reading
2	Corpus development	Guidelines for building a corpus, and a Dutch corpus for lip reading
3	Data parametrization	Suitable methods for lip reading and processing the data
4	Experiments	Training and testing various lip readers
5	Analysis of results	Statistics on the performance of the different lip readers
6	Design new experiments	Analysis of the results from other points of view
7	Conclusions	Final conclusions and future work

1.5 Outline of the Dissertation

The dissertation is structured in nine chapters including this introductory chapter and three appendixes. The knowledge is presented following a methodology-theory-methods-results-conclusions approach. The appendices give extra information which could not be given throughout the dissertation but which exemplifies or completes the discussions.

Chapter 2 (*Methodology, Definitions and the State of the Art*) starts by introducing the relevant methodologies we used in the current research. The chapter goes further by introducing all the concepts needed to understand the technologies, experiments, results and discussions presented in this dissertation. This chapter is, therefore, the base chapter of the dissertation and in our opinion makes the starting point for reading this book. The chapter follows the entire process of building a lip reader from the preparation stage to the interpretation of the results. In the last section of this chapter we present a succinct list of the state of the art in lip reading at the time of writing this doctoral dissertation.

Chapter 3 (*Computational Models*) gives the theoretical foundation of the techniques used during our research. Firstly, we introduce the theoretical aspects of the inference engine used, namely the Hidden Markov Models. Secondly, each in its own section, we present the theoretical aspects and some practical usage information on three techniques used for data parametrization. The strength and weaknesses of each technique are analysed and real examples are given.

Chapter 4 (*Data Acquisition*) introduces our work on building a data corpus for the Dutch language to be used for lip reading and other connected applications. We start by analysing the problems of the existing data corpora and enumerate a set of requirements a data corpus needs to satisfy. This resulted in an incipient set of guidelines for building a data corpus for lip reading. The resulting corpus is the largest data corpus for lip reading in Dutch to the date of this writing. In this chapter we present in detail all the steps we took to compile the corpus, and analyse the difficulties we faced and present the solutions we have chosen. The chapter also includes statistics on the resulting corpus and ideas for future development. The analyses presented in this chapter were published in:

[**Chi07a**] Alin G. Chiţu and Leon J. M. Rothkrantz. Building a Data Corpus for Audio-Visual Speech Recognition. *In Proceedings of Euromedia2007*, pages 88–92, 2007;

[**Chi08a**] Alin G. Chiţu and Leon J. M. Rothkrantz. Dutch Multimodal Corpus for Speech Recognition. *In Proceedings of Workshop on Multimodal Corpora LREC 2008*, pages 56–59, 2008;

[**Chi09a**] Alin G. Chiţu and Leon J. M. Rothkrantz. The New Delft University of Technology Data Corpus for Audio-Visual Speech Recognition. *In Proceedings of Euromedia2009*, pages 63–69, Eurosis, April 2009.

The next three chapters introduce the three approaches we used for data parametrization. Each chapter is organized into three parts: introduction of the method (algorithm and implementation), empirical validation of the resulting feature vectors and analyses of the performance of the lip reader built on the corresponding feature type. While interpreting the results of each lip reader, across these chapters we also introduce various techniques we used in order to boost the performance of the resulting systems.

Chapter 5 (*Statistical Lip Geometry Estimation for Lip Reading*) introduces the first data parametrization method we used in the current research. Based on a statistical interpretation of the result of a colour based image filter, this method is special because it does not use any a-priory model knowledge. The performance of the lip readers built for various tasks is given and analysed. We introduce the results gradually, presenting the steps we took in order to improve the results. In the next chapters we skip this gradual approach and only present actions and analyses specific to the given methods.

Chapter 6 (*Active Appearance Models for Lip Reading*) introduces a top down approach to the parametrization of the input data, suitable for lip reading, based on the Active Appearance Models (AAM). In contrast with the previous approach, AAM makes use of an a-priory designated model. Therefore, its applicability is limited to physical systems of which geometry deformation is confined to a non-degenerative range. The advantage of this method is its accuracy and speed; with a well trained model and a good initialisation this method obtains high accuracy in real time.

Chapter 7 (*Optical Flow Analysis for Lip Reading*) introduces the use of optical flow analysis as information source for automatic lip reading. The features computed based on optical flow capture real motion information. This contrasts with the other approaches which recover the motion information by computing the first derivatives of the static features. We used this approach in order to answer the question related with the importance of directly recovering the motion on the speaker's face.

Chapter 8 (*Further Analysis of the Results and Other Experiments*) analyses the results of the lip readers built during our experiments from different points of view. We analyse, therefore, the influence of the data parametrization approach, the influence of the sampling rate of the video data, the performance of the lip readers as a function of the recognition task but also the performance as a function of the size of the corpus used. Many of the research questions are dealt with in this chapter. The analyses presented in this chapter are based both on experiments introduced in previous chapters but also on new experiments. This is the reason the analyses are presented in a different chapter.

The methods, the results and the analyses presented in the last four chapters were published in various international journals and conferences listed below:

[**Chi07c**] Alin G. Chiţu, Leon J. M. Rothkrantz, Pascal Wiggers, and Jacek C. Wojdel. Comparison between different feature extraction techniques for audio-visual speech recognition. *In Journal on Multimodal User Interfaces*, vol. 1, no. 1, pages 7–20, Springer, March 2007;

[**Chi07b**] Alin G. Chiţu and Leon J. M. Rothkrantz. The Influence of Video Sampling Rate on Lipreading Performance. *In The 12-th International Conference on Speech and Computer (SPECOM'2007)*, pages 678–684, Moscow State Linguistic University, Moscow, October 2007;

[**Chi09b**] Alin G. Chiţu and Leon J. M. Rothkrantz. Visual Speech recognition - Automatic System for Lip Reading of Dutch. *In Journal on Information Technologies and Control*, vol. year vii, no. 3, pages 2–9, Simolini-94, Sofia, Bulgaria, 2009;

[**accepted for publication in September 2010**] Alin G. Chiţu, Karin Driel, and Leon J. M. Rothkrantz. Automatic Lip Reading in the Dutch Language Using Active Appearance Models on High Speed Recordings, *In Proceedings of Text, Speech and Dialogue(TSD2010)*, Springer's LNCS, Lecture Notes in Computer Science, LNAI subseries, September 2010.

Chapter 9 (*Conclusions: Summing Up, General Thoughts and Future Directions*) summarizes the most important findings presented across this dissertation. It also presents our thoughts on the lip reading future development and concludes with our hope for a great leap in the performance of lip reading.

To aid the reader in achieving a complete image of the process of building a lip reader, we included extra information in the appendices at the end.

Appendix A gives a compact description of the major existing data corpora. We include in this table information on the purpose of each corpus and on its availability.

Appendix B gives exact description of the grammars used for the different recognition tasks.

Appendix C introduces the utterance types recorded in the NDUTAVSC corpus.

Appendix D shows an example of the utterances recorded for the NDUTAVSC corpus.

Chapter 2

Methodology, Definitions and the State of the Art

This chapter will shortly describe the process of building an automatic lip reader and will define and elaborate on the important concepts that constitute it. The role of this chapter is to make the rest of the dissertation easy to read. Therefore, the reader should find here a short introduction to all the concepts needed to read and understand the rest of the book. The computational models and techniques are introduced in this chapter as well. However, only their usage is covered here, the theoretical aspects and the strengths and limitations of each of the different techniques used in our research will be covered in Chapter 3. In the last section of this chapter we will present the state of the art in lip reading at the time of this research.

2.1 Building an Automatic Lip Reader: Overview

Building a lip reader is in many ways similar to building any automatic system which performs an autonomous role in its environment. The first decision needed to be made before starting the construction of the system is with respect to the role of the system and with respect to the environment where the system will be deployed. After establishing, in pattern recognition jargon, the recognition task, building the system consists of four separate stages: data acquisition, data parametrization, model training and model testing. Figure 2.1 describes the general process of building a lip reader. These activities are performed in cycles, the larger the cycle the less frequent its corresponding process is performed.

The data acquisition process should ensure that the resulting corpus correctly describes the distribution of the possible states of the modelled process. The importance of the data parametrization is twofold; it should extract only the relevant information from the data and it should reduce the dimensionality of the feature space, therefore increasing the tractability of the problem. Training and testing are

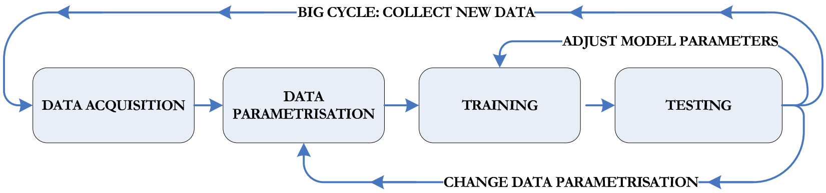


Figure 2.1: *The activity sequence for building a lip reader.*

dependent on the mathematical models chosen for inference. These range from plain heuristics to complex probabilistic graphical models. The training process should solve two problems: identify the structure of the models such as the number of parameters and their relation, and compute the values of the models' parameters. Training and testing is usually performed in a cycle which will fine tune the structure of the models and the values of the weights in the model. However, the data parametrization step is the one that is most of the time investigated, since there are many ways to extract suitable information for the process under study. Choosing the right parametrization is not straightforward and usually a trial and error sequence of experiments is started.

A lip reader and in general a speech recogniser is built for a particular target language. The recognition task, namely the size of the vocabulary and the type of utterances accepted, are paramount for the entire design of the system. For instance if for a small vocabulary (i.e. a few tens of words) one model can be used to recognise one entire word, for larger vocabularies it is more suitable to build sub-word models, i.e., to directly recognise sub-words and build the words and sentences using dictionaries and grammar networks.

So far, the most successful approach for speech recognition, and therefore also applied to lip reading, is the Bayesian approach. In the Bayesian approach, the recognition problem can be formulated as follows: given a set of possible words and an observation sequence $O = (O_1, O_2, ..., O_n)$ the solution of the recognition problem is the word that maximizes the probability $P(W|O)$. Based on the Bayesian rule we can write:

$$P(W|O) = \frac{P(O|W)P(W)}{P(O)}, \tag{2.1}$$

where $P(O|W)$ is the likelihood of the observation given the word W and $P(W)$, usually called the language model, represents the probability of the word W. The problem can be thus rewritten as:

$$\widehat{W} = \underset{W}{\operatorname{argmax}}(P(O|W)P(W)), \tag{2.2}$$

where $\widehat{W}$ is the recognized word. In the above equation the denominator $P(O)$ has been deleted since it does not influence the solution. Therefore, the recognition problem is reduced to building a language model $P(W)$ and a word model $P(O|W)$ for each legal word.

2.2 Recognition Task

Every system has a desired applicability. The recognition task defines for a speech recognizer[1] in general, and a lip reader in particular, the required settings in which the system is expected to function. The recognition task defines the environment in which the systems will be deployed, the characteristics of the users that the system is expected to handle and, very important for speech recognition, the types of utterances that should be recognized. The recognition task has a strong impact on the decisions made during the development stages of the system. As we already mentioned, the data corpus is tailored to capture the most relevant aspects for the recognition task. Also the basic recognition units are chosen to optimize the recognition process.

The environment can be either a controlled, noise free environment (e.g. office like environment where both the acoustics and the visual environment are controlled) or “uncontrolled” environment where the level of noise (i.e. acoustic and visual) has no a-priory set limit and is usually expected to be large (e.g. public spaces, the interior of a vehicle, crowded rooms). It is paramount to take into account the deployment environment at the training stage of the system. The corpus used for training should match as much as possible the conditions at the recognition place. In the case of lip reading the acoustic noise has no influence since it is only the visual information that is used. Therefore, in this case what is important is the illumination (e.g. diffuse light versus directional light, combination of natural light with artificial light, the temperature of the light, the position of the light, the shadows, etc.), the background (e.g. the colours in the background, the amount of movement in the background, the number of people in the background, etc.) and occlusions (e.g. the probability to have occlusions; for instance the position of the speaker with respect to the camera can give rise to occlusion by other entities but also to self occlusion).

The system can be multi-user, therefore, speaker independent or single user and thus speaker dependent. Usually, a multi-user system is adapted to a speaker dependent condition. The adaptation only requires a short corpus, while the deep training is performed on a large data corpus. This option is used by commercial ASR systems since it is not acceptable that the user is required to record a large corpus prior to using the system.

The most important aspect, the utterance types, is defined by the used vocabulary, the accepted sentences and the speaking style expected to be used. The legal sentences and the vocabulary are usually greatly correlated. For instance a system for recognition of telephone numbers, or in general of digit strings, has as legal sentences any fixed or variable length string of digits. The vocabulary in this situation consists of the digits “0” to “9”. A special case, and somewhat artificial because it has limited application, is the recognition of single digits. The same situation is found for the similar case of letter string recognition. The tasks introduced un-

[1] In this section, a speech recognizer (and speech recognition) is thought of as a multi-modal system that can recognize what is being said by fusing any number of sources of information, including audio, video, EEG, etc. In general, in the literature associated with our domain and also in this thesis, a speech recognizer (and speech recognition) means an aural speech recognizer (aural speech recognition), if not otherwise stated. More exactly the name of Automatic Speech Recognition (ASR) is used.

til now are the easiest tasks in speech recognition domain. Of similar complexity is the recognition of a set of words in isolation. The complexity in this case increases with the number of words in the vocabulary. The next step, with respect to complexity, is the case when the utterances recognized consist of a set of sentences that obey a strict grammar (e.g. a fixed set of commands with a very concise area of applicability). The last step on the complexity scale is as "continuous speech". The definition of continuous speech is somewhat under discussion. Some studies define continuous speech as any utterance that contains a string of valid words. By this definition the digit, letter and random word strings fit as continuous speech. However, in this thesis we define continuous speech as only those utterances that consist of valid sentences in the target language. This means that the grammar based recognition task can also be considered as continuous speech. The case of grammar based recognition task is, therefore, the easiest case of continuous speech task and for relatively small (i.e. less than 100 words) to medium vocabularies (i.e. a few thousand words) the aural speech recognition problem in controlled environments is solved. The current development in aural speech recognition deals with very large vocabularies (i.e. hundred thousand words) in controlled environments, for the so called "dictation" tasks. At the moment of the writing of this thesis there are commercial audio based speech recognition systems available which claim to have 99% accuracy for dictation tasks in a controlled environment. However, the "Holy Grail" of automatic speech recognition is still far from being achieved. The Holy Grail of speech recognition is the recognition of continuous and natural speech. By natural speech is meant utterances that may not be valid from the point of view of the grammatical rules of the target language, but that are perceived as natural by a human interlocutor. For instance hesitations, repeated words or intersections are common in regular conversations. These make the spoken utterance less clear and increase the complexity of the recognition task. Another aspect of the speech recognition domain's "Holy Grail" is building recognition systems that maintain the level of performance irrespective of the environment in which the system is deployed. This is actually the place where lip reading comes into action, by concurring to the enhancement of audio based speech recognition in noisy environments. Only after achieving these last two goals we could say that the automatic speech recognition has equalled human performance. One remark, related to this competition, is that even though the automatic systems outperform in general the humans in signal processing tasks and, therefore, achieve better scores for short utterances, for real life applications the gap is still enormous in favour of humans. One of the explanations seems to be that humans acquire with great ease the context of the conversation [Wig08].

As mentioned before, based on the recognition task we can decide what the basic recognition units should be. In the case of a reduced vocabulary it might be more suitable to use as basic recognition units the words in the vocabulary. Therefore, each word has associated a recognition module. This approach can be used in the case of digit strings and letter strings tasks. However, in the case of a larger vocabulary the number of models will be too large so a sub-word approach is preferred. In our work we used the sub-word approach for all recognition tasks.

From the point of view of the vocabulary and legal sentences, we defined five different recognition tasks which were reflected in the types of utterances we recorded in our corpus and on which we directed the analysis in our experiments. In the analysis and results sections we use the following codes to refer to them: digit strings (CD), letter strings (CL), grammar based utterances (GU), random continuous speech sentences (RS) and random utterances(all). Appendix B lists the grammars we used to generate the utterances in our corpus.

2.3 Data Corpus

A good data corpus should be well designed, should capture both the general and the particular aspects of a certain process. As in the case of speech recognition, training a lip reader requires much data and the quality of the data greatly influences the performance of the result. Whenever video data is considered, the amount of space necessary to save the data also increases considerably. For instance in [Mes98] the authors estimate that for training a multi-modality system for person identification the amount of data required is in order of TBytes (1000GBytes). The data acquisition process should ensure that the resulting corpus thoroughly covers the variability in the process being modelled. In general the data corpus should tightly cover the possible answers to the following questions:

1. What is the task of the system? Whatever task the system is built for the data corpus should contain sufficient data to cover that task.

2. What are the particularities of the environment where the system will be deployed? The corpus should contain support for the particular environment of the system. This could mean either to give support for adaptation or to sufficiently cover the given environment.

3. Who are the possible users of the system? The corpus should contain all the particularities of the possible users of the system.

The basic criterion of the corpus is to be large enough to successfully train the system. The performance of the system is, however, proportional with the degree of overlap between the corpus and the answers to the previous three questions.

For lip reading the data corpus consists of video recordings of speakers uttering a series of task related utterances. The field of view focuses on the lower half of the speaker face, since the lips are the information conveying elements. The speaker is instructed on the speech style required such as: speech rate, spelling versus normal speech, whispering versus normal pitch, etc.

However, partly because the field is still young, or partly because the time and resources necessary to record a visual data corpus can be overwhelming, the number of existing lip reading data corpora is still small compared with the number of audio datasets.

The next section evaluates the limitations of the existing corpora and gives a set of ground requirements for a data corpus.

2.3.1 On Building a Data Corpus for Lip Reading: A Comparison

In order to evaluate the results of different solutions to a certain problem, the data corpora used should be shared between researchers or otherwise there should exist a set of guidelines for building a corpus that all datasets should comply with. In the case when a data corpus is build with the intention to be made public, a greater level of reusability is required. In all cases, the first and probably the most important step in building a data corpus is to carefully state the targeted applications of the system that will be trained using the dataset.

Some of the most cited data corpora for lip reading are:
TULIPS1 [Mov95], AVletters [Mat96], AVOZES [GÖ4], CUAVE [Pat02], DAVID [Chi96], ViaVoice [Net00], DUTAVSC [Woj02], AVICAR [Lee04], AT&T [Pot97], CMU [Zha02], XM2VTSDB [Mes99], M2VTS [Pig97] and LIUM-AVS [Dau03]. With the exception of M2VTS which is in French, XM2VTSDB which is in four languages and DUTAVSC which is in Dutch the rest are only in English. Since the target language for our research was Dutch, we had only one option, namely the DUTAVSC (Delft University of Technology Audio-Visual Speech Corpus). For reasons that will be explained in the next paragraphs, we decided to build our own data corpus. This corpus was build as an extension to the DUTAVSC and is called NDUTAVSC [Chi09a] which stands for "New Delft University of Technology Audio-Visual Speech Corpus". This corpus will be introduced with great detail in Section 4.

From the point of view of speech recognition, the common limitations of a data corpus are:

1. The recordings contain only a small number of respondents. This greatly reduces the power of generalisation of the results, since it generally generates highly biased systems. The bias comes from the fact that not all the possible sources of variances are captured. The corpus is, therefore, unbalanced with respect to particularities of the user, such as facial particularities (e.g. moustache, beard and glasses), gender, age, race, dialect, education level, etc. This information could be used for building a context of the speaker which can be extremely valuable for building context aware speech recognisers [Wig08]. There are very few exceptions when the number of respondents is larger than 100. Even in these situations a good practice is to carefully record the speakers data, such as age, gender, race, dialect, etc.

2. The pool of utterances is usually very limited. The datasets usually contain only the digits or the letters of the Latin alphabet. An exception is the ViaVoice and DUTAVSC which contain also continuous speech. This induces a poor coverage of the set of phonemes and visemes in the language. If continuous speech is targeted then the prompts used should always contain phonetically rich words and sentences. Since the utterances contain only isolated words (letters or digits) the co-articulation effects are very poorly covered. As a good practice the language corpus should efficiently cover the possible combinations of phonemes in the language. This will help keeping the respondents effort in reasonable limits. Moreover, phonetically rich speech will also better represent the co-articulatory effect in the language.

3. The quality of the recordings is often very poor. This usually holds for the video data. It can be argued that for specific applications, such as speech recognition while driving a car, using dedicated databases (for instance AVICAR database) might better represent the specificity of the speech in the given situation. However, the use of such dataset will be entirely restricted to the application for which it is created.

4. The datasets are not publicly available. In general, many datasets that are described in scientific papers are not open to the scientific community. This makes it impossible to validate the results of the different methods. The comparison of the different methods is restricted. This is the case, for instance, for the ViaVoice data corpus which is a proprietary corpus.

One of the first datasets used for lip reading was TULIPS1. This database was assembled in 1995 and consists of very short recordings of 12 subjects uttering the first 4 digits in English. Another similar, very small dataset is AVletters. As the name suggests only the letters of the Latin alphabet are recorded. In this section we will underline the main issues of the existing datasets for speech recognition with respect to the audio and video quality and with respect to the language coverage. The dataset DUTAVSC will also be included here for comparison. Appendix A lists the main characteristics of the existing data corpora for lip reading.

Audio Quality

The complexity of audio data recording is much smaller than of the video recordings. Therefore, all datasets store the audio signal with sufficiently high accuracy, namely using a sample rate of 22 kHz to 48 kHz and a sample size of 16 bits. Therefore, the quality of the audio data is not subject to storage accuracy but from the perspective of recording environment. There are two approaches to the recordings environment: specific and neutral. In the first case the database is built with a very narrow application domain in mind such as speech recognition in the car. In this case the recording environment matches the conditions of the environment where the final system will be deployed. This approach can guarantee that the particularities of the target environment are closely matched. The downside of this approach is that the resulting corpus is too much dedicated to the problem domain and suffers from over training, and offers little generalization. In the second approach the dataset can be recorded in controlled, noise free environment. The advantage of this approach is the possibility to adapt the corpus to a specific environment in a post process. Therefore, a data corpus of this kind can be used for virtually any number of applications. The specific noise can be simulated or recorded in the required conditions and later superimposed on the clear audio data. An example of such a database is NOISEX-92[Var93]. This dataset contains white noise, pink noise, speech babble, factory noise, car interior noise, etc.

Video Quality

In the case of video data recording there is a larger number of important factors that control the success of the resulting data corpus. Hence, not only the environment, but also the equipment used for recording and other settings is actively influencing the final result. In the case of the environment the classification made for audio holds for video as well. The environment where the recordings are made is important since it can determine the illumination of the scene, and the background of the speakers. In the case of a controlled environment the speakers background is usually monochrome so that by using a "colour keying" technique the speaker can be placed in different locations inducing in this way some degree of visual noise. However, the illumination conditions of different environment are not as easily applied to the clean recordings, since the 3D information is not available anymore. In controlled environments the light is reflected by special panels which cast the light uniformly, reducing the artefacts on the speaker's faces.

Contrary to the audio case, the equipment used for recording plays a major role, because the resolution and the sample rate is still a heavy burden. Hence, the resolution of the recordings ranges from 100x75 pixels in Tulips1 and 80x60 pixels in AVletters datasets to 720x576 pixels in AVOZES and CUAVE datasets. The same improvement in quality is also observed in colour fidelity. The days of greyscale, 8bits per pixel images are long over, all new datasets are recorded in RGB 24 bits per pixel. This is very important because the discriminatory information is highly degraded by conversion to greyscale.

As shown in Section 2.3.2 another important factor is the frame rate at which the video recording is saved. The frame rate of the existing data corpora is conforming to one of the colour encoding systems used in broadcast television systems. Therefore, the video is recorded at 24Hz, 25Hz, 29.97Hz of 30Hz depending on the place in the world where the recordings are made. The data corpus used for the current research was recorded at 100Hz. The reasons for this choice will be discussed in Section 2.3.2.

The Region Of Interest (ROI) is important as well. For lip reading only the lower half of the face is important. However, in case context information is required, a larger area might be needed. Most of the datasets show, however, a passport like image of the speaker. In our opinion, at least for increasing the performance of the parametrization process a smaller ROI is more advantageous. Of course a ROI that is too narrow adds high constraints on the performances of the video camera used and it might be argued that this is not the case in real life where the resulting system will be used. Recording only the mouth area as is done in the Tulips1 data set is a tough goal to achieve in an uncontrolled environment. However, by using a face detection algorithm combined with a face tracking algorithm we could automatically focus and zoom in on the face of the speaker. A small ROI facilitates acquiring a much greater detail of the area of interest, in our case the mouth area, while keeping the resolution and, therefore, the bandwidth needs in manageable limits. Figure 2.2 shows some examples from six available data corpora.

The differences among the examples in this figure are clearly visible, with the exception of the DUTAVSC corpus, all other corpora reserve a small number of pixels for the mouth area. Table 2.1 gives the sizes of the mouth bounding box in all six samples.

This low level of detail makes the detection and tracking of the lips much more difficult. Any parametrization that considers a description of the shape of the mouth will be heavily influenced by image degradation. In the paper [Pot98] the authors re-

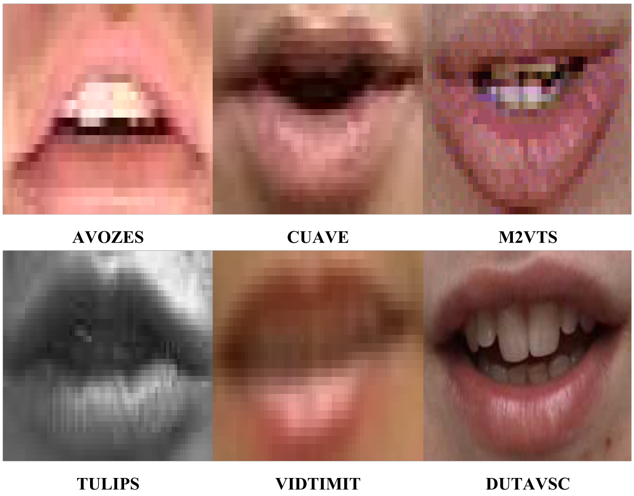


Figure 2.2: *The resolution of the ROI in some data corpora available for lip reading.*

Table 2.1: *Resolution of the mouth area in six known corpora for lip reading.*

Data corpus	Width	Height
AVOZES	122	24
CUAVE	75	34
M2VTS	46	28
TULIPS1	76	37
VidTIMIT	53	25
DUTAVSC	225	133

port that the degradation of the video signal by the image compression algorithm by the addition of white noise does not influence the lip reading performance unless the Signal to Noise Ratio(SNR) falls under some threshold: 50% and 15%, respectively. These findings are reported when the features used are a linear transformation of the intensities in the images, namely discrete wavelet transform.

Language Quality

By its nature lip reading requires, irrespective of the other qualities, that the data corpus has a good coverage of the language and task vocabulary. Therefore, in the case of a word based recognizer all the words in the vocabulary need to be present in the corpus. In the case of a sub-word recognizer every sub-word item needs to be present in the corpus in all existing contexts. Therefore, the co-articulatory effects appear with a reasonable frequency. However, due to the amount of work necessary and the storage and bandwidth required most of the data corpora only consider small recognition tasks and small language corpus. Most frequently the data corpora focus on the digits and letters of the language considered. These are recorded either isolated, or in short sequences, or as in DUTAVSC in spelling of words. Some corpora even only consider nonsense combinations of vowels(V) and consonants(C) (e.g. DAVID considers VCVCVC sequences, AVOZES repetitions of "ba" and "eo" constructions, AT&T CVC sequences). The continuous speech case is only considered in AVOZES which contains only 3 phonetically balanced sentences, in AVICAR which contains ten sentences from the TIMIT[Gar88] speech data corpus, XM2VTSDB and M2VTS which contains one random sentence and DUTAVSC which contains 80 phonetically rich sentences. The DUTAVSC is by far the most rich data corpus. The NDUTAVSC corpus which was built as an extension of DUTAVSC contains more than 2000 unique rich sentences. However, none of the existing corpora can match the language coverage offered by the data corpora used for speech recognition which can easily have a vocabulary of 100k words (e.g. the Polyphone corpus [Boo94] contains more than one million words recorded and a vocabulary of 150k words). The exact composition of the NDUTAVSC corpus is included in Chapter 4.

2.3.2 High Speed Recordings

As mentioned in Section 1.3, the first application of lip reading is to enhance the performance of a speech recognizer. One option to fuse the information from the two modalities to generate a common audio-visual speech recognizer is to concatenate the features extracted and train a single common model. This approach is called early fusion or feature fusion and is a popular approach in the lip reading community [Wig02].

The speech is usually recorded at a rate in the range 22 kHz to 48 kHz and the speech features are computed using a Hamming window of 30ms with 10 ms of overlapping on each side. Therefore, the audio modality is sampled at a 100Hz rate. On the other side the video modality can only provide 24 to 30 samples per second in standard setting. This implies that in the case of the early fusion paradigm an interpolation of the visual feature stream is needed to be able to synchronize the two modalities. Therefore, the question that arises is what is the influence of the recording frame rate on the performances of the recognition system?

In their paper [Wil98a] from 1998, Williams and his colleagues analysed this question from the human point of view. They devised experiments to analyse the influence of the frame rate on the ability of the human subjects. During the analysis they studied in what degree the confusion matrices of visemes are affected by changing the frame rate. The frame rates considered where 30Hz, 15Hz, 10Hz, 5Hz, and 2Hz respectively. The main finding was that a minimum frame rate of 10Hz is necessary to maintain the viseme grouping (i.e. the phoneme groups that define the visemes).

Potamianos et al. analysed this question from the point of view of a lip reader [Pot98]. The authors trained a lip reader using the same features but extracted at different frame rates, namely, 60, 30, 20, 15, 12, 10 and some other smaller steps. This experiment concludes that the lower limit for acceptable lip reading performance is 15 Hz.

The above results seem to answer the question about the influence of the frame rate on the lip reading performance. The conclusion seems to be that the lower limit for the recording frame rate that keeps the performance of lip reading in acceptable limits is 15Hz. Moreover, the conclusion of the authors of [Pot98] was that any larger frame rate has marginal benefits to the performance of the lip reader.

However, in both papers discussed above the data corpora used for analysis were, in our opinion, not completely adequate to answer the question of the influence of frame rate on lip reading performance from the point of view of an automatic lip reader. In the first paper the subjects were presented small CV combinations, while in the second paper the corpus used for analysis consisted of strings of four digits. Not only the corpus consisted of utterances that were too short to be used but there was neither indication nor control on the speech rate used to produce the corpus items.

The measure for speech rate is *Words Per Minute*(WPM). WPM is in fact a measure for input and output speed when human communication is considered. It is used, therefore, to characterize keyboard typing, handwriting, reading and comprehension but also listening and speaking. The maximum speech rate recorded to date is 595WPM. In general it is considered that 150-160WPM is the range that people feel comfortable with ([Wil98b]). As the speech rate passes the 200WPM threshold the average human listener already has difficulties following the message. The paper [Omo99] shows that an adult listener can maintain full comprehension at 210WPS only when the speech is compressed. In order to account for the length of the words when computing the speech rate (i.e. longer words should have more weight than shorter words when computing the number of words per minute) a word is usually defined with respect to multiples of 5 letters (e.g. “five” accounts for a word, while “multiples” accounts for two words). This has a large impact when analysing the speech rate for the Dutch language since this language uses compound word constructs for new concepts and, therefore, the words tend to be very long (e.g. “privébankrekening” (private bank account) is one word in Dutch).

To check the influence of the speech rate on the recorded data we considered the data corpus DUTAVSC which is recorded at standard 25Hz. We first checked the distribution of the WPM in this database. Subsequently, we analysed the distribution of the amount of data per unit of visual speech: visemes. Eventually, we analysed on a number of datasets the amount of information lost when the frame rate is lowered. The results of this analysis were published in [Chi07b]. The speech rate found in the DUTAVSC corpus ranges between 23WPM to 231WPM with a mean of 108WMP and a standard deviation of 36WPM. When 5-letter words are considered we found the speech rate between 24WPM to 277WPM with a mean of 137WPM and a standard deviation of 57WPM. The next step was to analyse the amount of data per viseme in the DUTAVSC corpus. Figure 2.3 shows the histogram of the number of frames per viseme in the DUTAVSC corpus. A strong pattern is visible in this image, namely there are utterances that are clustered in the range 2 to 4 and utterances that spread almost uniformly. Therefore, the conclusion is that there is a large difference with respect to data coverage of the visemes among the utterances in the corpus. As a general measure the mean number of frames per viseme found on the entire data corpus was 6 with a standard error of 5. Figure 2.5 shows the case of faster speech rate and Figure 2.4 shows the situation for lower speech rate. The mean number of frames per viseme was 3 for the high speech rate and 11 for the low speech rate; therefore, there is a significant difference between the two cases. Checking the utterance we found that in the low speech rate set we have only spellings, phone and account number utterances, while in the high speech rate set we have continuous speech utterances. It is important to mention here that the creators of the DUTAVSC corpus did not intent to analyse high speech rate and low speech rate and, therefore, did not instruct the speakers to make a difference in their speech rate. Therefore, all speakers spoke with their natural speech rhythm. However, as shown by the WPM values, the speech rate varies naturally with the task the speakers have. For instance the mean WPM obtained in the set of connected digits and letters and spellings utterances was 70WPM, while in the set of continuous speech set it was 128WPM. The same remark should be made when considering the results in the paper [Chi07b]. The conclusion in this paper was that 15Hz rate is too low to reliably capture natural speech in general, which is consistent with the findings in the studies discussed above. However, when high speech rate is considered going to 25Hz frame rate does still not provide enough data to assure a good coverage of the visemes. A frame rate higher than 250Hz is definitely an overkill. The conclusion we arrive at is that in our endeavour towards a robust lip reader we should investigate more carefully the impact of speech rate on the performance of an automatic lip reader. Therefore, we decided to use a high speed camera for our recordings and record with a 100Hz rate.

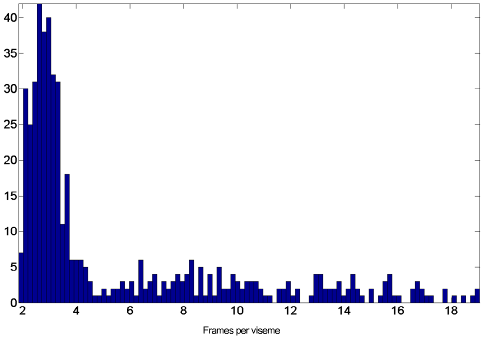


Figure 2.3: *The viseme coverage by visual data in the DUTAVSC corpus. All utterances are included in the histogram. The video stream is recorded at 25Hz rate.*

2.3.3 Data Annotation Versus Data Parametrization

Sometimes there is confusion between data annotation and data parametrization concepts. In the present research we use data annotation during data corpus development. For each utterance in the corpus, we automatically recorded the corresponding recognition task, the speech style, the recording session and the take number and we manually recorded any issues such as mispronunciation or other particularities found during analysis. During the annotation process we also made sure that only the corrected transcription is saved in the corpus. During this process

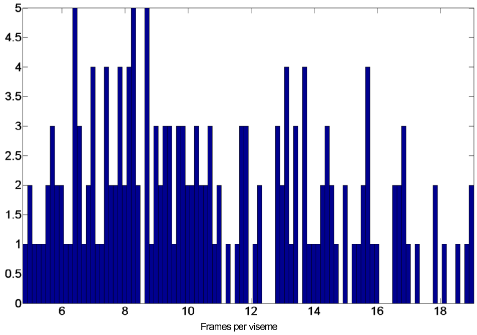


Figure 2.4: *The viseme coverage by visual data in the DUTAVSC corpus. Only the low speech utterances are included in the histogram, namely the spellings and connected digit utterances. The video stream is recorded at 25Hz rate.*

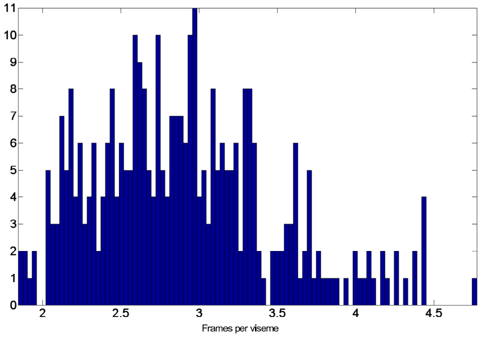


Figure 2.5: *The viseme coverage by visual data in the DUTAVSC corpus. Only the utterances that contain continuous speech are included in the histogram. The video stream is recorded at 25Hz rate.*

all the recordings were auditioned and checked for consistency from both acoustic and visual perspectives. Data parametrization is the process of extracting the suitable information for lip reading on which the recognition system will be realized. The results of the data parametrization are called features or feature vectors. Since a lip reading system is a classic example of a system that requires the use of *supervised learning* techniques both processes are equally important for the success of the system. Carefully pre-processing and organizing the data is, therefore, important if we want to avoid the well recognized phrase: *Garbage In, Garbage Out!*

2.3.4 Conclusions

The quality of the data corpus used has a great impact on the results of the research initiative. For this reason, a good data corpus should be built following some strict rules that guarantee the success of the final product. Also due to various limiting factors, the data corpora should exactly state the recognition tasks for which it was created. Since our research was focused on lip reading in the Dutch language we had only one data corpus available. Even though the DUTAVSC corpus is larger and the continuous speech task was better defined than in other corpora, it was still not sufficient for the goal of our research. Therefore, we decided to build a new corpus starting from the existing one. We used this approach because we wanted to be able to compare the previous results with the results obtained based on the new corpus. The new corpus not only is larger but it was recorded at high speed video rate and contains synchronized dual view, frontal and 90° profile. The existing corpora fail in many aspects; however, they were a good starting point for our work on building a data corpus. The protocol used during the recording sessions was written to avoid the issues found in the corpora investigated.

2.4 Data Parametrization

The row recorded video data as such is not suitable for lip reading. The first problem with such an approach is the resulting large dimension of the problem. With the current computational power solving problems in such high dimensional space is not viable. The second problem is that the row data inherits all sources of correlations many of which are not linked with speech production. Therefore, the row data is correlated with the settings of the camera, with the recording environment such as illumination conditions, and also with the affective state of the speaker and the identity of the speaker. To exemplify we can compare the lip reading task with the person identification and emotion recognition tasks. What represents valuable information for the person identification task or for emotion recognition task is in the case of lip reading a source of noise.

Therefore, the purpose of data parametrization is to extract the most relevant features with respect to lip reading from the row data while keeping the dimensionality of the problem as low as possible.

2.4.1 Feature Vectors Definition

There are many approaches to data parametrization, but with respect to the feature vectors definition they all fit in three broad classes: texture based features, geometric based features, and combination of texture and geometric features. A good overview of most of the feature extraction methods can be found in [Pot04].

In the first class the feature vectors are composed of pixels' intensities values or a transformation of them in some smaller feature space. The main function of the projection is to reduce the dimensionality of the feature space while preserving as much as possible the most relevant speech related information. *Principal Component Analysis*(PCA) is one of the first choices, and therefore very popular, and was used in many studies (e.g. [Bre93; Bre94; Duc94; Li95; Tom96; Chi97; Gra97; Li97; Lue97; Pot98; Dup00; Hon06]). The feature definition is based on the notion of *eigenfaces* or *eigenlips* which represent the eigenvectors of the training sets. An alternative to PCA, very common as well, is *Discrete Cosine Transform* (DCT) such as in [Duc95; Pr05; Hon06; Luc06]. *Linear Discriminant Analysis* (LDA), *Maximum Likelihood Data Rotation* (MLLT), *Discrete Wavelet Transform*, *Discrete Walsh Transform*

([Pot98]) are other methods that fit in this class and were used for lip reading. Virtually, any other method, usually borrowed from the data compression domain, which results in a lower dimensionality of the feature vectors can be applied for data parametrization in the lip reading domain. *Local Binary Patterns* (LBP) is just another technique, borrowed from the texture segmentation domain, and shows promising results for lip reading as well ([Mor07; Zha07; Kri08]). LBP was developed by Timo Ojala and Matti Pietikainen and presented in [Oja97]. A special place in this class is taken by the feature vectors that are based on *Optical Flow Analysis* (OFA) [Mas91; Mar95; Gra97; Fle00; Iwa01; Tam02; Fur03; Yos03; Yos04; Tam04; Chi07c; Chi09b] The optical flow is defined as *"the apparent velocity field in an image"*. This definition closely matches the affirmation of Bregler and Konig in their 1994 paper [Bre94]: *"The real information in lipreading lies in the temporal change of lip positions, rather than the absolute lip shape"*. The OFA can be used as well as a measure of the overall movement and be employed for onset/offset detection. The main advantage of this approach is that it can be easily automated, since it requires only the definition of the *Region Of Interest* (ROI). The ROI can be considered the bounding box of the face or the bounding box of the mouth, thus requiring some object detection and tracking algorithm. A good example is the face detection algorithm developed by Viola and Jones in [Vio01]. The main disadvantage of this type of features is that the a-priory information about lip reading is not inherently used in the process of feature extraction. Therefore, there is minimum control over the information contained in the resulting feature vectors, on whether this information is relevant for lip reading or not. The exceptions can be the OPA and LBP where the analysis is usually performed in carefully chosen regions around the mouth. Section 3.3 gives a detailed description of the OFA problem and presents some of the most successful solutions. This section introduces the mathematical foundations of the optical flow and presents several solutions to the optical flow problem. Chapter 7 presents the set of features we defined based on OFA and analyzes the performance of the lip reading system trained using the OFA based features on our data corpus.

The features from the second class share the belief that in order to accurately capture the most relevant features, with respect to lip reading, a careful description of the contour of the speaker's mouth is needed. The feature extraction proceeds in two steps; first a number of key points are detected and based on these points the mouth contour is recovered, and second the feature vectors are defined based on the shape of the mouth. The detection of the key points is performed based on colour segmentation techniques that identify pixels that are on the lips. Thereafter, the contour of the lips is usually extracted by imposing a lip model to the detected points. These methods are using the so called "smart snakes" ([Lie99; Lue97; Sal07]), or as called in [Eve04] "jumping snakes", or later on *Active Shape Models* (ASM)([Lue96; Pr05; Mor07] or *Active Contour Models* (ACM). Any other parametric model can be used here. The lips' contour is usually detected as a result of an iterative process which searches to minimise the error between the real contour and the approximation of the contour the parametric model allows for. The actual feature vectors are defined in the second step. The feature vectors fall into two categories here: model based features and mouth high level features. In the first category the feature vectors contain directly the parameters of the models used for describing the mouth contour. In the second category the feature vectors contain measurable quantities, which are meaningful to humans. The most used high level features are *mouth height*, *mouth width*, *contour perimeter*, *aperture height*, *aperture width*, *aperture area*, *mouth area*, *aperture angle* and other relations among these (e.g. the ratio between the width and the height) ([Chi09b; GÖ0b; GÖ0a; Kum07; Mat02; Yos04]). In Chapter 5 we introduce the *Statistical Lip Geometry Estimation* (LGE), a feature extraction method developed by Wojdel and Rothkrantz ([Woj00]) that we used in our research. The results obtained based on feature vectors computed with this method are also given in this chapter. This method is special because it is a model free approach for describing the shape of the lips. It strongly depends, however, on the performance of the image segmentation technique used to detect the pixels which belong to the lips.

The third class consists of feature vectors that contain both geometric and texture features. The features from each category are usually concatenated in a larger feature vector. For instance [Dup00] and [Lue96] combine ASM with PCA features and [Chi97] combines snake features with PCA. It was shown that the tongue, teeth and cavity have great influence on lip reading ([Wil98a]), therefore, the addition of these appearance related elements has significant influence on the performance of lip reading ([Chi07c]). A special example is the so called *Active Appearance Models* (AAM) ([Coo98]) which combines the ASM method with texture based information to accurately detect the shape of the mouth or the face. The searching algorithm is iteratively adjusting the shape such that to minimise the error between the generated shape and the real shape. The core of AAM is PCA which is applied three times, on the shape space, on the texture space and on the combined space of shape and texture. The AAM based features can either consist of AAM model parameters in which case we have a combined geometric and texture feature vector, or of high level features computed based on the shape generated in which case we have a geometric feature vector. The AAM technique will be discussed in detailed in Section 3.4. The lip reading results based on high level feature vectors which are computed starting from the lips' shape generated based on AAM are given in Chapter 6.

2.4.2 Lip Reading Relevant Feature Space

It is important to assure that the features obtained from the data parametrization process contain only speech information and that the sources of information that constitute noise from the speech point of view are removed. However, this is not as easily done. From the point of view of lip reading, the sources of noise are: the person's appearance dependent information (i.e. the particularities of the speaker such as skin colour, lips appearance, the presence of moustache and/or beard, the speaking style), the speaker's emotional state (e.g. speaking and laughing can be very disturbing from the point of view of a lip reader), the recording environment (e.g. the illumination conditions can influence the feature values). The texture based features are more prone to include person and environment dependent information, and therefore are more suitable to person identification tasks than to lip reading. However, as we have seen in the previous section many researchers used them for lip reading because they are easy to implement and automate. On the other side the geometric features can also carry speaker dependent information related to the lips appearances (e.g. for instance the dynamic range of the mouth height is expected to vary with the speaker). In his thesis ([Woj03] in Chapter 4 Wojdel showed that the geometric feature vectors extracted from different speakers overlap to a larger degree than the raw data features. He proposed a Person Adaptive PCA approach, used for reducing the dimensionality of the feature space, to try to alleviate this problem.

The feature normalization is a process that is meant to calibrate the dynamic ranges of the features vectors. A direct result of normalization can also be the partial removal of the influence of the speaker and recording particularities, especially in the case of geometric features. For instance a common practice when computing high level features based on special points on the speaker's face is to search for a distance that is not changed during the entire recording. This distance is used as a unit distance for further computations. In the case of a full face description the distance between the speaker's eyes is used ([Kob92]) as "base" distance. Since in the case of lip reading only the lower half of the face is processed another base distance should be found. During our research we found that the size of the philtrum at its upper end (i.e. next to the nose base) remains virtually unchanged during speech. Therefore, it is a good candidate for the base distance. This is the distance we used to normalize the high level features based on AAM, which are introduced in Chapter 6.

2.4.3 Higher Order Features

The research of speech recognition has shown that the performance of a speech recognizer can be greatly improved by adding the time derivatives to the static features. Usually, only the first and the second order regression coefficients are used. These are the so called *Deltas* and *Accelerations* features. Less frequently the third order derivatives are used as well.

The first derivatives of the static features are computed using the centred finite difference approach given in [You05]. Hence the delta feature at time t was computed by the following regression formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \tag{2.3}$$

where Θ is the window size and $c_t \pm \theta$ are the neighbouring static features.

To cope with the difficulties encountered at the boundaries, the first and the last features need to be replicated, respectively. To obtain the second time derivatives the above formula is applied again to the delta features.

In the case of lip reading, and visual data processing in general, the features computed based on optical flow analysis can be considered as a type of high order features. The question is whether the motion information is expressed in these high order features and to what extent.

2.4.4 Image Segmentation Fundamentals

Lip reading deals with visual data which represents recordings of the speaker's face. In other words a lip reader's working data consists of strings of digital images. A digital image is a binary representation of a two-dimensional image which is saved in raster or vectorial form. Lip reading deals with raster images which consist of a finite collection of image elements called pixels organized as a two-dimensional array. The value of each pixel is an integer number corresponding to the colour of that pixel. This value can either be an index in a look up table, or the real mathematical coordinates in a certain colour space. There are several colour spaces defined for different applications. As Charles Poynton nicely observes by making a parallel with the cartography domain [Poy99], there is no single best colour space that satisfies the needs of all applications. All have their advantages and disadvantages with respect to the particular application. The data parametrization algorithms make use of various colour representations. Therefore, we consider that no discussion about data parametrization for lip reading is complete without an introduction to colour specification and colour spaces. This section defines the most important concepts used in colour technology and presents the most important colour spaces from the point of view of lip reading. A number of colour transformations and lips segmentation techniques are also presented.

Colour and Colour Related Concepts

Colour is the perceptual result of the interaction of light in the visible region of the spectrum, wavelengths in the range 380nm to 750nm, with the eye's photoreceptor cells and processed by the brain. Figure 2.6 shows a linear representation of the visible spectrum. Since this image shows pure spectral colours only a subset of the colours distinguished by the human brain are shown. Colours like pink and magenta are obtained by mixing light of various wavelengths.

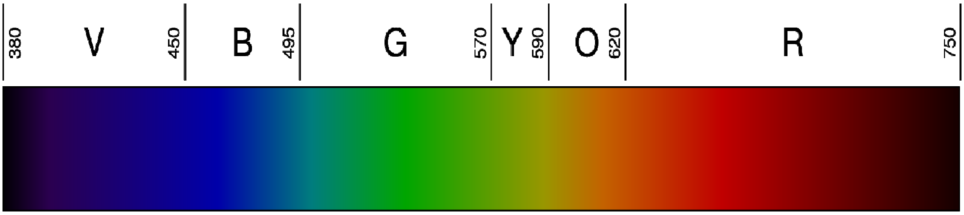


Figure 2.6: *A linear representation of the visible light spectrum.**

A human retina has three types of photoreceptive cells: cones, rods and photosensitive ganglion cells. However, only the cones are responsive for the creation of

[1] *This image is courtesy of www.wikipedia.org. The author has released it into the public domain.

the colour sensation in the brain. There are three types of photoreceptive cone cells, which have different spectral response curves. The cones are labelled according to the locations of the peaks of the corresponding curves: short (S), medium (M) and long (L). They are also classified as blue, green and red, respectively, even though their peaks are not exactly at the corresponding wavelengths. The most visible mismatch is in the case of the red cones which have a spectral response curve that peaks in the yellow-red area. The response of the human cones to monochromatic stimuli is shown in Figure 2.7. Figure 2.8 shows an aggregated image of the sensitivity of the human eye for pure spectral colour stimuli. As can be seen in this image, the sensitivity of the human eye is around the value 550 nanometres which corresponds to the green colour.

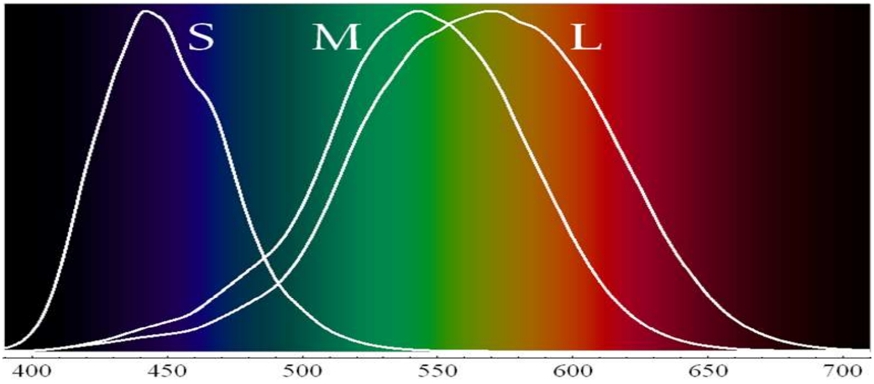


Figure 2.7: *Normalized response spectra of human cones, S, M, and L types, to monochromatic spectral stimuli, with wavelength given in nanometres.**

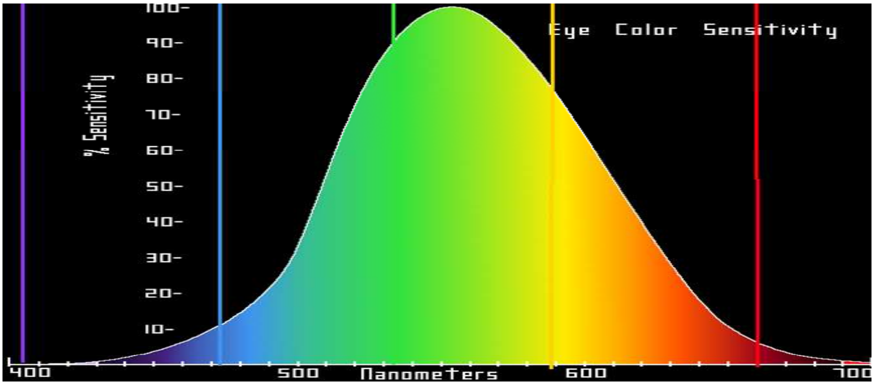


Figure 2.8: *Sensitivity diagram of the human eye to pure spectral colours.**

The conclusion here is that since there are exactly three colour photoreceptive cones, a three dimensional space is necessary and sufficient to describe a colour. Sir Isaac Newton said (cited in [Poy99]), *Indeed rays, properly expressed, are not* *coloured.* Therefore, colour exists only in our brain facilitated by the eyes. Moreover, it should be noted that the peak response of human colour receptors varies among individuals and that colour perception is not identical. Also the visible region is defined with respect to human's vision, since other species can have different visual affinities. For instance bees and other species of insects but also some birds and reptiles can see in the ultraviolet range well beyond human's perception.

The *Commission Internationale de L'Éclairage*(CIE) is the international authority on light, illumination, colour, and colour spaces and was established in 1931 and is based in Vienna, Austria.

Brightness is defined by the CIE as *the attribute of visual sensation according to which an area appears to emit more or less light.* Because brightness is a property that is not easily measurable, CIE introduced the notion of *luminance* which is defined as the measure of the luminous intensity per unit area of light travelling in a given direction. It describes the amount of light that passes through or is emitted from a particular area, and falls within a given solid angle and depends on the spectral sensitivity function which is characteristic to vision. The SI unit for luminance is candela per square metre (cd/m2). Usually, in the video domain, it is normalized with respect to a *reference white.*

Lightness is the perceptual response to luminance by the human vision. The human vision has a nonlinear perceptual response to brightness ([Poy99]).

Hue is defined by the CIE as *the attribute of a visual sensation according to which the area appears to be similar to one of the perceived colours, red, yellow, green and blue, or a combination of two of them.* In other words the hue is the notion we normally use in our everyday life when talking about colours.

Saturation is defined by the CIE as *the colourfulness of an area judged in proportion to its brightness.* Therefore, we can think of saturation as the level of purity of the perceived colour.

As concluded above, three components are necessary and sufficient to specify a colour. Therefore, the standards defined by CIE consist of series of methods to map the perceived colour to a vector with three components that are the mathematical coordinates of the colour space.

Colour Representation and Colour Spaces

There are many systems defined for colour specification. The most used today include CIE XYZ, CIE xyZ, CIE L*u*v*, CIE L*a*b*, CIE L*ch, HSV/HSB and HSL/HLS/HSI. For performance reasons the systems used for digital image coding are somewhat different from the colour specification systems. These include linear RGB, nonlinear gamma-corrected R'G'B', nonlinear CMY, Y'C_BC_R, HSV/HSB, Y'UV and Y'IQ. The colour space used in digital computer imaging is R'G'B' (see Figure 2.9) which is the RGB colour space after the *gamma correction* is applied.

Gamma correction is the process through which, in a video system the light intensity is transformed to a nonlinear video signal. This correction is necessary to correct for the non-proportionality between the voltage and the intensity of the light. The gamma correction is applied at the time of capture at the camera. It is important to note that the RGB colour space is merely a convenient means for representing colour, and is not necessarily directly based on the types of cones in the human eye.

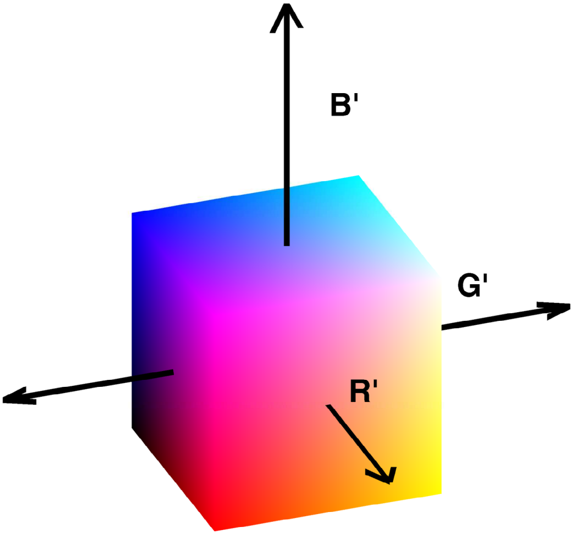


Figure 2.9: *R'G'B' Cube.*

One disadvantage of the RGB colour space is that it does not show perceptual uniformity, which means that an equal translation in any of the space directions has a different perceptual result depending on the initial location. The R'G'B' has somewhat better performance with respect to uniformity. The same disadvantage is also found to the CIE XYZ colour representation. The CIE needed more than a decade of research to find a new colour representation that did not show this problem. However, the research resulted in two colour specifications CIE L*u*v* and L*a*b*, without a clear inclination to one or the other. Since the two colour spaces were still not perfect there were successive refinements of their definitions. Another property of these colour spaces that was desired was to be similar to the human visual perception. This means that two colours that are perceived as similar by a human observer should lie in the same region of space; therefore, the Euclidian distance between them should be minimal.

Lip Pixel Segmentation

In order to describe the geometry of the lips, the first step is to detect which pixels belong to the lips. This process is called lips segmentation and is a type of image segmentation. The simplest way for doing this segmentation is to use a thresholding technique based on the appropriate colour encoding system. The process consists of filtering out the pixels which have a colour that is too far away from the desired colour, which is the colour of lips in our case. While a binary segmentation can be used as well, in general it is more important to generate a result in probabilistic terms. The result of the segmentation process should describe the degrees of *belongness* of a given pixel to the object, the lips in our case. Therefore, we use for our research a parabolic thresholding approach. This method for image segmentation was first proposed in [Coi96]. The result of the segmentation is computed according to the following parabolic shaped function:

$$F_X(x) = \begin{cases} 1 - \frac{(x-x_0)^2}{w^2}, & |x - x_0] \leq w \\ 0, & |x - x_0| > w \end{cases} . \tag{2.4}$$

Attention should be paid to the dynamic range of the variable x, for instance in the case of Hue-based filtering since Hue is a circular variable. The filter is defined by the range centre, x_0, and by its half width w. Both values should be calibrated in advance in order to obtain sufficient accuracy. Figure 2.10 shows an example of such a filter.

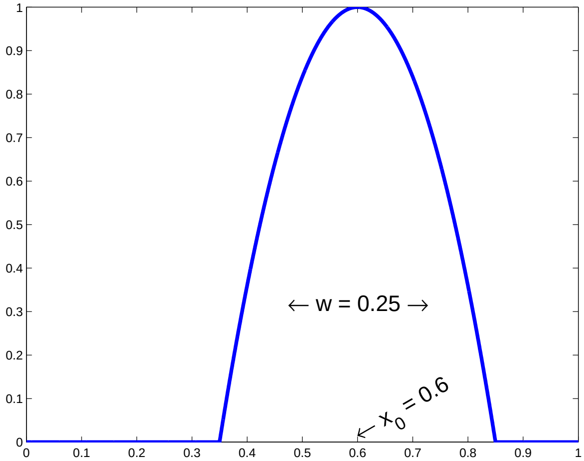


Figure 2.10: *A realisation of the parabolic filter for* $x_0 = 0.6$ *and* $w = 0.25$.

We may also combine a series of such parabolic shaped filters for more robust lip detection. Using the product of the Hue-based filter and a Value-based filter can for example remove some of the noise in the dark or bright areas of the image where the hue values behave rather randomly.

It is important, therefore, to use a uniform component of the colour representation. Another approach is to use other clustering techniques for pixels classification (e.g. neural networks as in [Woj03]). We found that the black-box approach based on a neural network outperformed the parabolic thresholding approach. However, the neural network approach is more computational intensive.

Most of the work in image segmentation is performed in the so called *greyscale* domain. The greyscale is a measure of brightness and therefore is related to *luma*(Y') channel as defined in the Y'UV colour spaces and to L* as defined in the L*a*b*. However, the greyscale is obtained from the R'G'B' by first applying the gamma expansion to obtain the RGB values then computing the luminance and then applying the gamma compression to the result. However, sometimes the greyscale is computed from R'G'B' by simply averaging the three values. The luminance and luma are also frequently used. In [Wan07] the authors use the combined spaces L*a*b* and L*u*v* for colour representation in order to segment the lips. For lip reading the hue as defined in the HSV colour space is very often used ([Zha00; Woj03] since the lips are very distinctive in this domain. Hulbert and Poggio ([Hul98]) observed that the difference between the red and the green channel in the R'G'B' space is greater for lips than for skin. Based on this fact they proposed the use of a transformation which maximizes this difference. The resulting quantity is called *pseudo hue* and is defined as follows:

$$H = \frac{R'}{G' + R'}. \tag{2.5}$$

In [Ozg08] the authors use the R'G'B', pseudo hue, hue and their combinations for lip segmentation. A comparison of different performances for human faces segmentation in difficult backgrounds is presented in [Ter99]. Based on the normalized pseudo-hue, Eveno et al. ([Eve01]) define a chromatic curve that is used to make the difference between the pixels on the lips and the face pixels even larger. They also introduce a correction to the R'G'B' components that is meant to reduce the dependence on the luminance, and therefore, make the models less vulnerable to illumination variability. The (R_c, G_c, B_c) values are computed as follows:

$$Ch_c(x,y) = \frac{Ch(x,y)}{Ch(x,y) + (b-a)L(x,y) + a}, \tag{2.6}$$

where Ch is one of the R', G' or B' channels, $L(x,y)$ is the luminance and the pair $(a,b) = (0.4, 0.5)$ are such chosen as to define the importance of the luminance.

2.5 Lip Reading Primitives

This section introduces the *visemes* which are the lip reading counterparts of the *phonemes*.

2.5.1 Phonemes

In any spoken language a phoneme is the smallest segmental unit of sound which generates a meaningful contrast between utterances. Thus a phoneme is a group of slightly different sounds which are all perceived to have the same function by speakers of the language or dialect in question. An example of a phoneme is the group of /p/ sounds in the words *pit spin* and *tip*. Even though these /p/ sounds are formed differently and are slightly different sounds they belong to the same phoneme in English because for an English speaker interchanging the sounds will not change the meaning of the word, however strange the word will sound. The phones, or sounds, that make up a phoneme are called allophones.

A speech recognizer can be built at word level or at sub-word level. While for a small vocabulary recognition task a word level system might be preferred, for large vocabulary, continuous speech task systems the phonemes are used as building blocks. Therefore, each phoneme in the target language corresponds to a recognition model in the speech recognizer.

In the Dutch language, approximately 40 distinguishable phonemes are defined. However, there can be slight differences among different phoneme and phoneme sets as a consequence of the target dialect and definition of accepted words. In the present research we used the phoneme set defined in [Dam94]. One problem is generated, for instance, by the neologisms. These words are divided in two classes: the ones that are already established into the language (e.g. the words of French origin) and have a stable pronunciation but which contain phonemes that are still under-represented in the language and a second class of very new words (e.g. the International English words from various technical and economical background) which bring a set of new phonemes that have no correspondence in Dutch. Tables 2.2 and 2.3 show the phonemes of the Dutch language as used in the Polyphone corpus. The phonemes are given in International Phonetic Alphabet(IPA), Speech Assessment Methods Phonetic Alphabet(SAMPA)[2], and HTK[3] notations, respectively.

2.5.2 From Phonemes to Visemes

Even though the definition of the concept of phoneme crosses the boundary of the auditory realm, and therefore is not bound to any sensory modality, the term "*viseme*" is used as the counter part of phoneme in the visual modality. The term was introduced by Fisher in [Fis68].

The visemes have a similar definition with the phonemes, namely, a viseme is a set of indistinguishable phonemes; indistinguishable phonemes from the point of view of the visual information available and not as in the phonemes case from the point of view of their meaning. There are two direct consequences of this definition. Firstly, there is no exact method of deciding the number and composition of the viseme classes; this is actually done either by a theoretical discussion of auditory-visual lip reading of phonemes or by modelling the human ability of recognizing the phonemes in the absence of the auditory stimulus, therefore, by modelling the degree of confusion of phonemes in the visual modality. Secondly, since there is no one-to-one mapping between the phonetic transcription of an utterance and the corresponding visual transcription, the separability of utterances in the visual modality decreases, which decreases the theoretical performance of a lip reader. The dependence of the visemes on the phonemes can be thought of as one reason why a new term was needed.

[2] The SAMPA notation is a computer readable phonetic notation based on IPA.

[3] HTK is a toolkit for building Hidden Markov Models (HMMs), developed at the Machine Intelligence Laboratory of the Cambridge University Engineering Department. For more information please visit http://htk.eng.cam.ac.uk/ or read "The HTK Book" [You05]. This column should be used to understand the results shown in the later chapters of the book.

Table 2.2: *Polyphone's Dutch phoneme set: consonants.*

	Symbol			Example Word		
	IPA	SAMPA	HTK	Orthography	Transcription	Translation
1	p	p	p	pak	p a k	package
2	b	b	b	bak	b a k	container
3	t	t	t	tak	t a k	branch
4	d	d	d	dak	d a k	roof
5	k	k	k	kat	k a t	cat
6	g	g	gg	goal	gg oo l	goal(sports)
7	f	f	f	fel	f e l	fierce
8	v	v	v	vel	v e l	sheet
9	s	s	s	sein	s ei n	signal
10	z	z	z	zijn	z ei n	to be
11	x	x	x	acht	a x t	eight
12	ɣ	G	g	negen	n ee g at	nine
13	ɦ	h	h	hand	h a n t	hand
14	ʒ	Z	zj	bagage	b a g aa zj at	luggage
15	ʃ	S	sh	sjaal	sh aa l	scarf
16	m	m	m	met	m e t	with
17	n	n	n	nek	n e k	neck
18	ŋ	N	nn	bang	b a nn	scared
19	l	l	l	land	l a n t	country
20	r	R	r	rand	r a n t	edge
21	ʋ	w	w	wit	w i t	white
22	j	j	j	ja	j a	yes

Unlike for English, to date there is only a limited number of publications which deal with the definition of visemes in Dutch; this is an almost complete list of them: [Bre85], [Cor84], [Egg64], [Son94], [Vis99] and [Beu96]. The papers [Son94] and [Beu96] (cited in [Woj03]) are the only examples, at least to the author's knowledge, where the classification of the viseme sets is done by elicitation of the human confusion matrices of phonemes. The authors of [Son94] found in their experiments that the Dutch lip readers are only able to recognize four consonantal and four vocalic visemes which are shown in Table 2.4. In Table 2.4 the separation between the *a)* sets and *b)* sets was done only by keener respondents.

Table 2.3: *Polyphone's Dutch phoneme set: vowels.*

	Symbol			Example Word		
	IPA	SAMPA	HTK	Orthography	Transcription	Translation
1	ɪ	I	i	kip	k i p	chicken
2	ɛ	E	e	pet	p e t	cap
3	ɑ	A	a	pad	p a t	path
4	ɔ	O	o	pot	p o t	pot
5	ʏ	Y	y	put	p y t	put
6	ə	@	at	de	d at	the
7	i	i	ie	vier	v ie r	four
8	y	y	yy	vuur	v yy r	fire
9	u	u	u	voer	v u r	to feed
10	aː	a:	aa	vaar	v aa r	sailing
11	eː	e:	ee	veer	v ee r	spring
12	øː	2:	eu	deur	d eu r	door
13	oː	o:	oo	door	d oo r	through
14	ɛi	Ei	ei	fijn	f ei n	fine
15	œ	9y	ui	huis	h ui s	house
16	ʌu	Au	ou	goud	x ou t	gold
17	ɛː	E:	eh	créme	k r eh m	cream
18	œː	9:	euh	freule	f r euh l at	madame
19	ɔː	O:	oh	roze	r oh z at	pink

Table 2.4: *Viseme sets according to Son et al. in SAMPA notation.*

		Viseme class	Description
a)	1	p, b, m	bi-labial consonants
	2	f, v, w	labiodental consonants
	3	s, z, S	non-labial front fricative
	4a	t, d, n, j, l	non-labial front consonants
	4b	k, R, x, N, h	non-labial back consonants
b)	5a	i, I, e:, E	close and half-close front vowels(unrounded)
	5b	Ei, a:, a	half-open and open vowels(unrounded)
	6	u, y, 9:, O	short back vowels (rounded)
	7	2:, o:	long back vowels (rounded)
	8	Au, 9y	closing and rounding diphthongs

Table 2.5 shows the viseme set as reported in [Vis99]. The difference in the viseme sets obtained by different researchers can be attributed to various factors such as the set of utterances used, the respondents or the clustering parameters used. For instance the authors of [Wil97] use a 75% minimum agreement to form a new cluster. Son et al. report in [Son94] that the level of proficiency in lip reading of the respondents do not influence the Dutch viseme classification.

Table 2.5: *Viseme sets according to Visser et al. in SAMPA notation.*

	Viseme class		Viseme class
1	p, b, m	8	I, e:
2	f, v, w	9	E, E:
3	s, z	10	A
4	S, Z	11	@
5	G, k, x, n, N, R, j	12	i
6	t, d	13	O, Y, y, u, 2:, o:, 9y, 9:, O:
7	l	14	a:

In the present research we use the viseme sets as shown in Table 2.6, which is a combination of the viseme sets found in the literature. The combination was made such that it agrees with the phoneme set used. It should be noted that while the Polyphone corpus uses 41 phonemes, in Table 2.4 appear only 34 phonemes and in Table 2.5 appear only 37 phonemes. For instance the phonemes *h* and *ei* did not appear in any of the above classifications, therefore, they were added as separate viseme classes. The last column in the table gives the working name for each class. This name will be used later in the results sections.

Table 2.6: *The viseme sets used in the current research in working notation.*

	Viseme class	Working notation
1	at	at
2	ie	ie
3	l	l
4	a	a
5	aa	aa
6	f, v, w	fvw
7	s, z	sz
8	sh, zj	shzj
9	p, b, m	pbm
10	g, k, x, n, nn, r, j	gkx
11	t, d	td
12	i, ee	iee
13	e, eh, uh	eeh
14	o, y, yy, u, eu, oo, euh, oh, ou	oyu
15	h	h
16	ei	ei

Figure 2.11 shows instances for each of the 16 visemes listed in Table 2.6, instances taken from the data corpus built for the current research. The images show only the apex of each viseme. Of course, the complete impression can only be acquired from watching the entire sequences of images. Also, since we did not record isolated visemes, with the exception of the letters which are transcribed with only one viseme (e.g. $< E >$=[iee]), that the co-articulation effect can be very large in

the resulting mouth movement. However, we can conclude that there is a large difference between the visemes and that a native Dutch speaker could easily recognize each viseme. It should be noted that the face appearance depends as well on the way the speaker talks and articulates the words, and therefore, it is not always easy to recognize all the visemes. Figure 2.12 shows four instances with the viseme [gkx] which has proved extremely difficult to be classified during our experiments. As we can see in Table 2.6 this viseme gathers a large set of phonemes that correspond all to non-labial back consonants, which means that they are produced inside and probably show little correlation with the position of the lips.

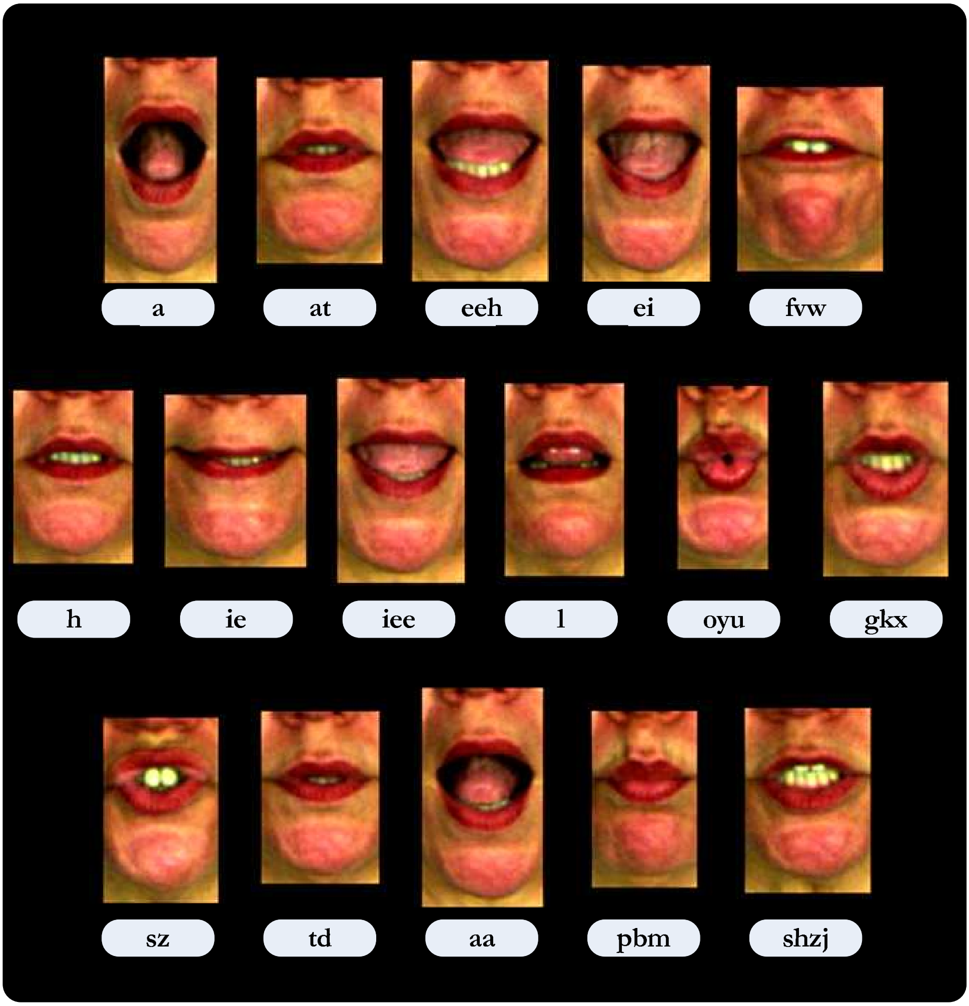


Figure 2.11: *Instances with all the visemes used in our research taken from NDUTAVSC corpus. From top-left to bottom-right we have: [a] as in word < acht > = [a gkx td], [at] as in word < zeven > = [sz iee fvw at], [eeh] as in letter < F > = [eeh fvw], [ei] as in letter < IJ > = [ei], [fvw] as in letter < F > = [eeh fvw], [h] as in letter < H > = [h aa], [ie] as in letter < I > = [ie], [iee] as in letter < E > = [iee], [l] as in letter < L > = [eeh l], [oyu] as in letter U = [oyu], [gkx] as in letter < Q > = [gkx oyu], [sz] as in letter < Z > = [sz eeh td], [td] as in letter < D > =[td iee], [aa] as in letter < A > = [aa], [pbm] as in letter < P > = [pbm iee], and [shzj] as in word < plaatsje > = [pbm l aa td shzj at].*

2.5.3 Separability of Utterances as a Result of Viseme Definition

As we mentioned in the previous section, the definition of visemes brings along a lower theoretical performance for a lip reader in comparison with its counter part in

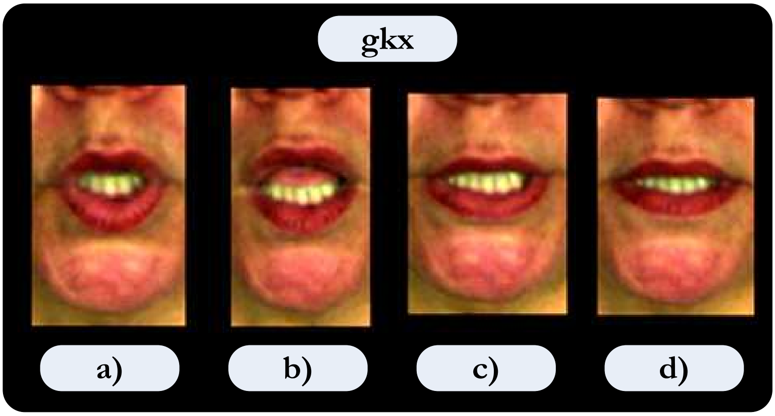


Figure 2.12: *Four instances of viseme [gkx] taken from the NDUTAVSC corpus. We have [gkx]: a) as in letter $< Q >$ = [gkx oyu], b) as in letter $< G >$ = [gkx iee], c) as in letter $< J >$ = [gkx iee], and d) as in letter $< K >$ = [gkx aa].*

the aural modality. In other words, the separability of the words in the vocabulary decreases because some words will end up with the same phonetic transcription after mapping to the visual modality. For instance, when mapping the 26 letters of the Dutch alphabet to the viseme transcription, we only obtain 20 distinct entities because the pairs (B, P), (D, T), (G, J), (N, R), (O, U) and (V, W) have the same viseme representation. A similar problem appears for the English alphabet where the pairs (A, K), (B, P), (C, Z), (D, T), (S, X), and (Q, U) are also found visually indistinguishable ([Pet88]). In his work, J. Wojdel [Woj03] introduces the notion of *Viseme Syllable Set* which is a more informative viseme classification which will differentiate the viseme set based also on their syllabic context. However, as the author remarks, the number of classes grows exponentially with the task vocabulary which makes them unsuitable for the implementation. Based on the amount of information the viseme sets carry, it was computed that for the Polyphone corpus the separability of the words can decrease in the range from 3%, when the full phonetic representation is used, to 13%, when only the viseme representation is used ([Woj03]). In the corpus we used there was a 7% decrease in separability. Therefore, the 13% decrease can be seen as a theoretical upper boundary of the performance of the lip reader. An additional decrease is caused by the propagation of the misclassifications, resulted from the more difficult task of correcting a misclassified word due to a larger set of close neighbours.

2.6 Hidden Markov Models Methodology

The theoretical consideration of the *Hidden Markov Models* (HMMs) are given in Chapter 3. In the current section we shortly introduce the reader to the concepts behind the HMMs and to the usage of HMMs as statistical inference engines. Thereafter, we define the basic elements needed to build a lip reader based on the HMMs approach and discuss some of the major decisions made at design time.

The HMMs were introduced in the late 1960s in a series of statistical papers by Leonard E. Baum and others ([Bau66; Bau67; Bau68; Bau70; Bau72]). After a period of partial obscurity, by the middle of 1970s the use of HMMs exploded in many areas were time-series were used. These models are based on the well known Markov process used in probability theory. A Markov process is a one step memory stochastic process. This process considers that the future state of the physical system that it models depends only on the current state of that system. This property is called Markov property. A process is, thus, seen as a finite state process with a probabilistic transition function. In a Markov model each state corresponds to one observable event, hence we can say that in the case of a Markov model we observe directly the state of the system. This is not appropriate for the real systems from two reasons: firstly it is seldom possible to actually observe the state of a system (in most cases we measure a quantity that has a functional relationship with the system we want to model), and secondly the large number of possible observations (i.e. states), will make the model enlarge exponentially. To overcome these problems the HMM introduces the notion of hidden states. Hence the states of the system are now hidden and the observations are made over some variables that are induced by the current state of the system.

To date, HMMs still provide the best performance in the speech reading domain. Other domains related with temporal pattern recognition are also using HMMs approach such as: gesture and body motion, optical character recognition, machine translation, bioinformatics and genomics. Their use in bioinformatics started in middle 1980s for problems such as: prediction of protein-coding regions in genome sequences, modelling families of related DNA, prediction of secondary structure elements from protein primary sequences, and so on. Since lip reading is still speech recognition, only in a different modality, HMMs are a logical choice. The use of HMMs is also justified by the fact that it easily gives support to the feature fusion of the aural and visual modalities for multi-modality speech recognition. Other derivations of the HMMs such as Cartesian Product HMM, Factorial HMM, Linked HMM, Hierarchical HMM, Coupled HMM, Multi-stream HMM, Product Multi-stream HMM can cope with other fusion approaches. However, even though these models aim at overcoming known problems with the classical HMMs, they exhibit an increased complexity (for instance the number of parameters to be estimated grows exponentially requiring an enlarged dataset for training) which greatly diminishes their use.

Since the generalization of the application of HMM, there are many implementations such as “The Dragon system” ([Bak75]), “The HARPY system” ([Low76]), and HTK Toolkit ([You05]) among many others. In our research we used regular HMMs and we used the implementation offered by the HTK Toolkit. The toolkit contains all the tools necessary for building a speech recognizer, from data acquisition to data preparation, from creating of the HMMs models to training the models and tuning them in a highly parameterised manner and to finally testing the recognizer. It also provides software for language modelling and for the building of other components needed such as dictionaries and word networks. The main advantage in working with HTK toolkit is its high modularity. Even though it was created specially for speech recognition, thanks to its modular architecture, it can be easily used for different problems such as in our case for lip reading. We only needed to implement our own feature extraction modules and simply replace the speech data with the visual data. The training of the models is done using the Baum-Welsh algorithm. Finally the testing of the system is done with the Viterbi algorithm.

2.6.1 Modelling the Visemes Using HMM

As a sub-word based speech recogniser, the building blocks of our lip reader are the visemes of the Dutch language. Therefore, one HMM corresponds to one viseme. To the set of visemes are added two special models, namely *sp* for "short pause" and *sil* for "silence". The *sp* model is used for recognition of the short pause between words, while *sil* is used for the silence moments before and after the utterance. Depending on the recognition task, some visemes do not appear at all in the expected utterances and are, therefore, excluded from the study. This is the case for the digit and letter tasks. The set of visemes which appear in the digit recognition task are listed in Table 2.7 and the set of visemes which appear in the letter recognition task are listed in Table 2.8. The visemes "*at*" and "*a*" are only present in the digit set, while the visemes "*aa*" and "*pbm*" are only present in the letter set.

Table 2.7: *The viseme set in HTK working notation for the digit recognition task.*

	Viseme		Viseme
1	gkx	8	ei
2	oyu	9	sz
3	l	10	eeh
4	iee	11	at
5	td	12	a
6	fvw	13	sil
7	ie	14	sp

Table 2.8: *The viseme set in HTK working notation for the letter recognition task.*

	Viseme		Viseme
1	aa	9	h
2	pbm	10	ie
3	iee	11	ei
4	sz	12	l
5	td	13	oyu
6	eeh	14	sil
7	fvw	15	sp
8	gkx		

The topology of the models used for modelling the visemes, usually used for phoneme-based speech recognition as well, is a 3-state left-right with no skips as shown in Figure 2.13. For implementation reasons, HTK requires that the models start and end with a non emitting node that facilitate the generation of recognition networks. A recognition network consists of a string of linked models which are used during recognition by matching to the input utterance. In Figure 2.13 the numbers on the arcs represent the initial transition probabilities, set before training. Under the emitting states there is a generic drawing of the distribution of the feature vectors which is approximated by a mixture of Gaussian distributions.

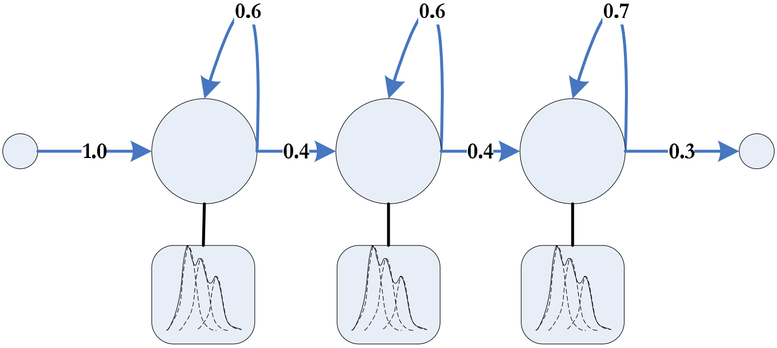


Figure 2.13: *The models used for modelling the visemes. The topology is 5-State Left-Right with three emitting states. The arcs are annotated with transition probabilities.*

The modelling of the two silence models are introduced in the next section.

2.6.2 Silence and Pause Models

It is not possible to build a continuous speech recognizer without including a model for silence. However, there are two types of silence, the ones between the words and the ones that appear in the beginning of the utterance and at the end of the utterance. The silence model that covers the entering and exit time of the utterances can be modelled using the same topology as for viseme models (i.e. 3-state left-right topology). However, in order to make the model more robust by allowing the states to absorb more non verbal mouth movement, the silence model is modified so that a backwards transition from state 4 to state 2 is accepted. The model for short pause is build starting from the model for [sil]. The short pause model is a so called *tee-model* and has a single emitting state which is tied to the central state of the [sp] model. This means that the central state of the [sil] model and the emitting state of the [sp] model share the same Gaussian mixture and therefore are trained using the same data. Parameter tying is very often used in speech recognition for the cases when there is not sufficient data for training models for similar entities. The topology used for the two silence models is shown in Figure 2.14.

The silence models defined above are the same as the ones used for speech recognition. However, there is a big difference between the concept of silence in speech recognition and the concept of silence in lip reading. Consequently, the noise can have a more robust definition. For instance, in the case of visual speech the speaker can move his mouth for non verbal reasons (e.g. to moisture his lips, or to exteriorise

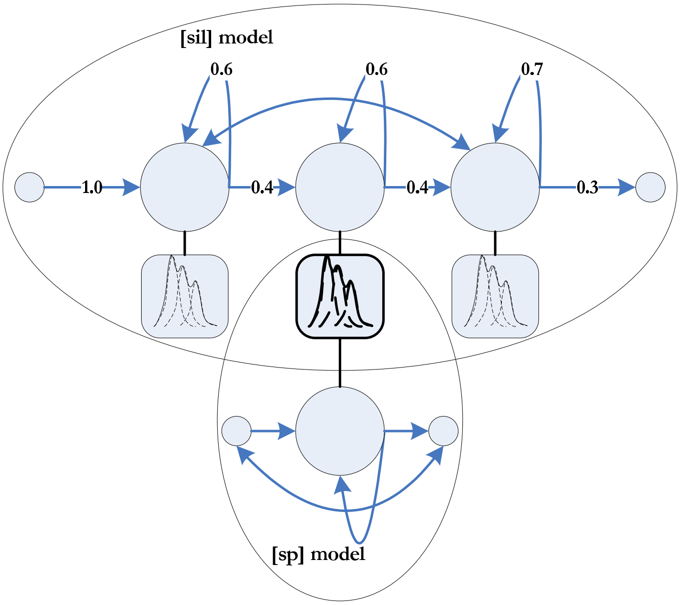


Figure 2.14: *The models used for modelling the silence.*

the emotional status by showing a facial expression). The noise sources are more diverse for lip reading. Even though the silence model has an extra backward arc which should, in principle, also accommodate for noise in the training data, we found out in our experiments that the silence model defined in this way did not perform at the same level as in the case of speech. As we will see later in the results sections, sometimes the insertion rate was unexpectedly large. This can also be due to poorly trained silence models.

2.6.3 Modelling the Low Level Context Using Tri-visemes

In order to model the context at the level of the visemes, each viseme is considered in all the possible contexts. Only a one step context is considered, namely for each viseme only the left and the right possible visemes are considered, therefore, the name of the new entity is tri-viseme. The notation for tri-visemes is lf-vis+rt, where “vis” is the viseme in question, “lf” is the left context and “rt” is right context. For instance the word *nul*[4] with the viseme transcription *gkx oyu l* will generate the following tri-visemes: *gkx+oyu*, *gkx-oyu+l* and *oyu-l*. The context of each viseme can be build at word level, also called *word internal*, or at the level of utterance called *word external*. In the first case, for finding all possible contexts of a viseme, only the words in the vocabulary are considered, while in the second case also the possible combinations of words can build the context. It should be noted that sometimes bi-visemes (i.e. viseme context containing only the left or the right viseme) are also

[4] The Dutch word for the number zero.

generated. For each tri-viseme, a new model will be build which makes the number of models explode, making the data requirements for training a tri-viseme based recognizer many times larger. The major problem with the tri-visemes is that some contexts can appear only once (or a very small number of times) in the training data, or can even be absent from the training data, as in the case of trans-word boundary contexts. To solve this problem the parameter tying technique is used. The clustering of possible similar contexts can be made either by a data-driven approach, or by the use of decision trees. Even after the parameter tying, there can still be tri-viseme models which are undertrained.

2.6.4 Gaussian Mixtures

The HMM approach considers that each of the emitting states in the model will be described by a continuous density distribution. This distribution is approximated in HTK by a mixture of Gaussian distributions. Building of the models in HTK starts by using only one Gaussian distribution. In the refining step the number of Gaussian mixtures is increased iteratively by 1 or 2 units until the optimum number of components is obtained. By monitoring the performance change, the optimum number of mixtures can be found. During our experiments we iteratively increased the number of mixtures by one until a maximum of 32 mixtures. The “magic” number 32 was found sufficiently big to cover the optimum number of mixtures in all the experiments.

2.7 Language Models

A speech recognizer in general, therefore a lip reader as well, needs a method to choose among the possible hypotheses build based on the acoustic (visual) data evidence. This method should not only be able to judge if a possible hypothesis is a legal sentence according to the rules of the given language, but also to decide which of the hypotheses is syntactically and semantically more suitable. This task is fulfilled by a language model which defines the set of legal sentences in a language and assigns a probability to them. This means in mathematical terms that a language model represents a probability distribution over the space of the words and the legal sequences build with them, which in equation 2.2 is denoted by $P(W)$.

The main problem of the language modelling is how to assign a probability at the sentence level. The most used approach is the chain rule from the probability theory that, based on the notion of conditional probabilities, permits splitting the sentence in smaller units for which assigning the probability is much easier:

$$P(W_1...W_n) = P(W_1)P(W_2|W_1)P(W_3|W_1W_2)...P(W_n|W_1W_2...W_{n-1}). \tag{2.7}$$

In a way, the above formula suggests that the main task of a language model will be to predict the likelihood of the next word based on the previous words in the sentence. Of course, it is also possible to directly consider the whole sentence as an indivisible entity. There are many ways to develop a language model but the most successful approaches are *n-grams* and *grammar based language models*. Other improvements and additions to these models try to make them more robust, by, for instance, splitting the sentences space into smaller classes and building language models for each class. A context based approach can also improve the performance of the language model. This means building language models for classes based on topic.

2.7.1 Grammar Based Language Models

The grammar based language models use a number of constraints on the set of sentences that are legal in the language. Therefore, the syntactic structure of the sentences controls almost entirely the distribution over the language. The grammar rules dictate, based on a given history, the most probable next word. There is a big correlation between the recognition task and the set of grammatical rules used to build the language model. For instance, in the case of the digit string recognition task, the length of the strings is reinforced through the grammar. The grammar can also link multiple recognition tasks (e.g. the single digit recognition task with digit string recognition tasks).

The task used for digit recognition in our case is given in the following listing. The digit recognition task consists in our experiments of either one digit utterances or exactly eight digit strings.

```
$digit = \<0\> | \<1\> | \<2\> | \<3\> | \<4\> | \<5\> | \<6\> |
         \<7\> | \<8\> | \<9\>;
$eightdigits = $digit $digit $digit $digit
               $digit $digit $digit $digit;
$connectedDigits = $digit | $eightdigits;

(
     SENT-START $connectedDigits SENT-END
)
```

The $ sign announces the declaration of a variable. The “|” represents a disjunctive gate, meaning that any of the operands can appear. The SENT-START and SENT-END are two special words which delimitate an utterance and translate to silence.

The grammars we used for our experiments to define the recognition tasks are included in Appendix B. The later recognition task includes digit strings, letter strings, topic based utterances where the chosen topic is banking applications and word strings.

For continuous speech task, the grammar based language model approach is not feasible. To handle this task we used an n-gram approach which is covered in the next section.

2.7.2 n-Grams

n-Grams are the most popular type of language models. An n-gram considers all sequences of (n-1) words as being in the same set of equivalence. The probability of the n$^{\text{th}}$ word depends only on the distribution of the words conditioned on the history's class of equivalence. Therefore, the solution of the n-gram is a counting problem, which makes building such a language model extremely easy to implement. However, as the length of the history increases, the number of n-grams increases exponentially requiring a very large amount of data in order to accurately describe the conditional distributions. Therefore, the length of the history (i.e. the number "n") is determined by the amount of data available. In most cases n is equal to 2, 3 or 4, but the tri-grams are by far the most preferred n-grams.

2.8 Measures for System Performance

For checking the performance of a lip reader we used the concept of *Word Correct Rate* (WCR) and *Accuracy* (Acc) as they are defined by the HTK Toolkit, the tool we used for building and testing our systems. The counter part of the Acc measure, namely the *Word Error Rate* (WER) is the most popular metric used for assessing the performance of speech recognition or machine translation systems. Both measures are usually computed and reported at both item and sequence levels. Even though a lip reader can be build starting from both word and sub-word levels, in general, the measurement of performance is done at word and word sequence levels, respectively.

The difficulty of evaluating the performance comes from the fact that the recognized sequence can have a different length from the reference sequence. The WER measure is derived from the Levenshtein distance also called the edit distance. The edit distance between two strings is the minimum number of insertions, deletions and substitutions required to transform one string into the other [Lev66]. The WER is the edit distance between the reference word sequence and the result of the automatic lip reader normalised by the length of the reference word sequence. This normalisation removes the dependence on the target string's length and makes possible the comparison between different systems and also different recognitions tasks. Therefore, WER is defined as follows:

$$WER = \frac{S + D + I}{N} \times 100\%, \tag{2.8}$$

where N is the total number of words in the reference sequence, D is the number of deleted words from the reference sequence, S is the number of substituted words in the resulting sequence and I is the number of inserted words in the resulting sequence (i.e. the words that do not exist in the reference sequence). The *Word Recognition Rate*(WRR) or Acc is thus defined as:

$$WRR = 100\% - WER = (1 - \frac{S + D + I}{N}) \times 100\% = \frac{H - I}{N} \times 100\%, \tag{2.9}$$

where $H = N - D - S$ and represents the number of correctly recognized words. Based on H we define the WCR as follow:

$$WCR = \frac{H}{N} \times 100\%. \tag{2.10}$$

As noted in [McC05] the WRR suffers from a number of limitations which sometimes makes it difficult to be interpreted. Due to the use of number of insertions in its definition, the WRR is allowed to have negative values and is not bound to unity. On the other hand the WCR does not have this problem and therefore has a clearer interpretation. Also, the WRR does not consider the varying importance of the words, therefore further limiting its analytical power. In order to correct some of these problems, other performance measures were developed. For instance, the paper [McC05] introduces a measure for evaluating the system results based on an information retrieval problem approach, where each word is one unit of information. The measure is based on the concepts of recall and precision. The paper [Mor04] introduces a new measure *Word Information Preserved* (WIP). This is derived as an approximate measure of the mutual information between the reference and the target sequence and is defined by:

$$WIP = \frac{H^2}{(H+S+D)(H+S+I)}. \tag{2.11}$$

In order to take into account the effect that different types of errors have on the result, [Hun89] introduced a weighted WER defined as:

$$WER = \frac{S + 0.5D + 0.5I}{N},$$

therefore given to the errors by substitution a larger weight than to the error by deletion or insertion.

Another measure for the goodness of a speech recognizer in general was introduced by Roger K. Moore [Moo77], HENR. The performance of the system is compared to human performance scores. While this measure is relatively independent of the test vocabulary and gives a comparison to the human performances, it has the disadvantage that it is very laborious to compute.

2.9 State of the Art in Lip Reading

It is about three decades since automatic lip reading domain emerged in the scientific community. However, only starting from the 90s, and more sustained in the second half of the 90s, the subject started to become viable. Even today it still lags the speech recognition by some decades. Until some years ago the most impeding factor was the computational power of the computers. Nowadays it is the difficulty in finding the most suitable visual features that capture the information related with what is being spoken. Also it is the hard problem of accurately detecting and tracking the facial elements that convey speech related information. The automatic and robust detection and tracking of the face elements is still not entirely achieved by the current technology. As in other similar visual pattern recognition applications, the two monsters “illumination variations” and “occlusions” are still alive and menacing. A special case of occlusion is in this case generated by the posture of the speaker. Therefore, any study concerning lip reading deals with the overwhelming task of manually or in the best case semi-automatically processing the data corpus. The data corpora for lip reading are still very small due to partially the storage and bandwidth limitations and other recording related settings, but much more limiting due to the overwhelming task of processing and preparing the data for experiments. Because of these issues, as presented in Section 2.3, each data corpus is created for a stated recognition task. The lip reading experiments to this date are limited to isolated or connected random words, isolated or connected digits, isolated or connected letters. Some of the reported performance is listed below. However, it is very important to keep in mind that, because the data corpus used influences in great respect the performance of the lip reader, a comparison among the experiments is not always possible. When the corpora are about the same, then the comparison of the different feature types and feature extraction techniques becomes feasible. It can still give an impression on the state of the art in lip reading.

The task of isolated letters was among the first analysed by Petajan et al. ([Pet88]) back in 1998. The authors report the correct recognition close to 90%. However, based on the AVletters data corpus, Matthews et al. ([Mat96]) reports only a 50% recognition rate. Li et al. ([Li95]) reports a perfect recognition 100% on the same task, but two years later in [Li97] only 90% recognition. The second most popular task is digit recognition either in isolation or as connected strings. Based on the TULIPS1 data corpus, which only contains the first four digits, Luettin et al. ([Lue96]) and Luettin and Thacker ([Lue97]) reported 83.3% and 88.5% recognition rates, respectively. Arsic and Thiran ([Ars06]) report on the same data corpus 81.25% and 89.6% depending on the feature extraction method. Other experiments with the digit recognition task are: Potamianos et al. ([Pot98] reported 95.7%, Dupont and Luettin ([Dup00]) reported 59.7%, Wojdel reported in his thesis ([Woj03]) 91.1% correct recognition and 81.1% accuracy, Patamianos et al. ([Pot04]) reported 63% and Pérez et al. ([Pr05]) 47%. Lucey and Potamianos ([Luc06]) reported 74.6% recognition rate for the isolated digits task. Potamianos et al. (Potamianos1998a) report 64.5% recognition rate for the connected letter task. For the isolated word task Nefian et al. ([Nef02]) report 66.9%, Zhang et al ([Zha02]) report 42%, Kumar et al. ([Kum07]) report 42.3%. We can conclude that there is still a large variation in the performances obtained, and there is still no convergence visible since the newer studies do not necessarily show an increase in accuracy. This is, to our opinion, clearly a sign of the immaturity of the lip reading domain. Also, as can be observed in the listing above, there are yet no results of experiments with continuous speech as defined in Section 2.2. Patamianos et al. ([Pot04]) report an extremely low result on the continuous speech task, namely 12%.

The lip reading domain is still young and there are many limiting factors that need to be conquered. Therefore, the experiments in lip reading are still dealing with relatively easy tasks. However, the promising results in these tasks give us hopes that larger experiments are possible. As the domain becomes more popular, the number of data corpora will increase and with a better cooperation among scientists it will be possible to better compare the achievements. However, as shown in Section 2.5.3 there are objective factors which limit the performance of the lip readers. Nevertheless, as shown in many studies, lip reading can be successfully used in conjunction with speech for an enhanced speech recognition system.

Chapter 3

Computational Models

This chapter introduces the theoretical aspects of the models, methods and algorithms used in the experiments presented in this thesis. Most of the concepts presented in this chapter were already introduced in Chapter 2. This chapter is intended for readers that want a detailed view on the mathematical foundations of the methodology used. This chapter is, therefore, not required for the understanding of the arguments, the experiments and the results presented in the later chapters. The chapter is divided into two parts: inference techniques and data parameterisations techniques.

3.1 Hidden Markov Models

This section introduces the hidden Markov models paradigm and gives the formal mathematical foundations to using this technique. The basic questions around the HMMs and the solution to them are briefly repeated here for the sake of completeness.

There are many books and tutorials written on this topic, especially from the point of view of HMMs use in speech recognition, but maybe the most complete and easy to follow tutorial is written by Lawrence R. Rabiner in 1988 ([Rab89]). At the time of writing of this thesis the paper [Rab89] was already cited in more than 10500 scientific papers.

The HMMs were introduced in the late 1960s in a series of statistical papers by Leonard E. Baum and others ([Bau66; Bau67; Bau68; Bau70; Bau72]). After a period of partial obscurity, by the middle of 1970s the use of HMMs exploded in many areas were time-series were used. Their use in speech recognition applications was a big step forward, and to date they still provide the best results in some applications, speech recognition included.

The HMMs are an extension of the well known Markov processes used in probability theory and were introduced as “probabilistic functions of Markov chains”. Therefore, the HMMs represent a stochastic approach in modelling the world. The main property of a Markov process, called "the Markov Property", is the notion of memory. A Markov chain has a finite memory which means that the future state of the physical system that it models depends only on the current state and a finite number consecutive past states of that system. The HMMs are actually based on a one step memory Markov process. This process considers that the future state of the physical system that it models depends only on the current state of that system. In mathematical terms we can have the following formalism. Consider a system that at any time can be in one of the N distinct states $S_1, S_2, ...S_N$, as shown in Figure 3.1. At any time step $t = 1, 2, ...$ the system can be in only one state which is denoted by q_t. The Markov property, in the case of a first order Markov chain, as is the case here, is expressed as follows:

$$P(q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_k, ...) = P(q_t = S_i | q_{t-1} = S_j). \tag{3.1}$$

Making the assumption that the Markov property is independent of the current time, namely

$$P(q_t = S_i | q_{t-1} = S_j) = P(q_2 = S_i | q_1 = S_j),\ t \geq 2, \tag{3.2}$$

the above probability is called the state transition probability and is denoted by a_{ij}:

$$a_{ij} = P(q_t = S_i | q_{t-1} = S_j),\ 1 \leq i, j \leq N. \tag{3.3}$$

The state transition probabilities have the standard stochastic properties, namely:

$$a_{ij} \geq 0,\ 1 \leq i, j \leq N,\ and \tag{3.4}$$

$$\sum_{j=1}^{N} a_{ij} = 1. \tag{3.5}$$

The assumption is that we can directly observe in which state the system is at any given moment in time. Such a model is called observable process and each state corresponds to one observable event. This is not appropriate for the real systems from two reasons: firstly it is seldom possible to actually observe the state of a system (in most cases we measure a quantity that has a probabilistic function over the state of the physical system), and secondly the large number of possible observations (i.e. states), will make the model to enlarge exponentially. Therefore, the actual state of the system is not observable, thus hidden, and can only be assessed through a set of observations produced by another stochastic process. In Figure 3.1 the round nodes represent the hidden states, and the rectangular nodes represent the visible stochastic processes that produce the observations. Usually, these observable nodes are omitted when representing a HMM, hence its graphical description coincides with the graphical description of a Markov process. The missing arcs in the Markov process correspond to a null probability.

To conclude, a HMM is characterized by:

1. The number of states N.

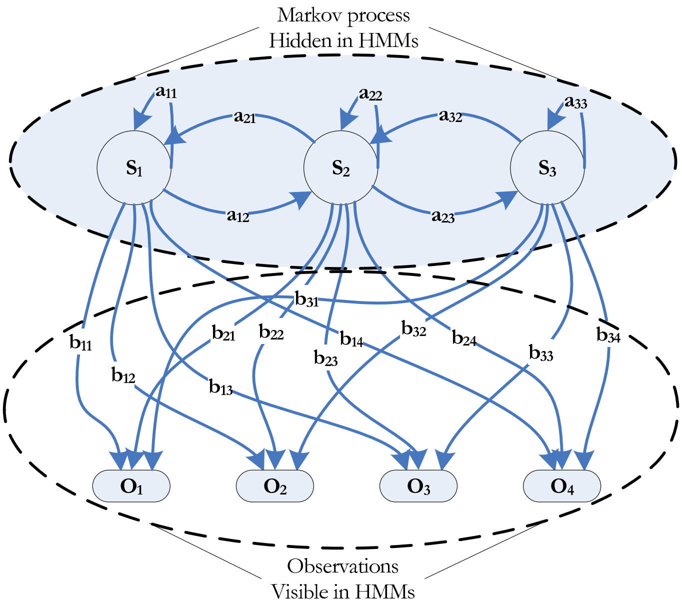


Figure 3.1: *A HMM with 3 hidden states (S_1, S_2, S_3) and 4 observation nodes (O_1, O_2, O_3, O_4). The transition probabilities are denoted by a_{ij} and the observation probabilities are denoted by $b_i(j) = b_{ij}$.*

2. The set of possible states $S = \{S_1, S_2, ..., S_N\}$.

3. The number of distinct observation symbols M, which form the output alphabet.

4. The individual symbols in the output alphabet $O = \{O_1, O_2, ..., O_M\}$.

5. The state transition probability coefficients matrix $A = (a_{ij})^{i,j}$, where

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i),\ 1 \leq i, j \leq N, \tag{3.6}$$

and

$$a_{ij} \geq 0,\ 1 \leq i, j \leq N,\ \textit{and} \sum_{j=1}^{N} a_{ij} = 1. \tag{3.7}$$

6. The probability distribution on the observation symbols in state j, denoted by $B = \{b_j(k)\}$, where

$$b_j(k) = P(O_t = O_k | q_t = S_j),\quad 1 \leq j \leq N,\quad 1 \leq k \leq M. \tag{3.8}$$

7. The initial state distribution $\pi = \{\pi_i\}$, where

$$\pi_j = P(q_1 = S_j), \quad 1 \leq j \leq N. \tag{3.9}$$

Written in a concise form, an HMM is entirely defined by the seven entities $\lambda = (N, S, M, O, A, B, \pi)$. For convenience, usually, the HMM is denoted only by the probabilistic distributions since they are of greater importance and make the difference between models with the same structure but modelling a different physical model, $\lambda = (A, B, \pi)$.

The immediate application of an HMM starting from the above definition is to use it as a generator of observation strings. This only requires sampling from the distributions that appear in the definition and, based on the result, traversing the states of the model and generating observations. However, the question that is of interest from the point of view of real applications is:

Given the observation sequence $O = O_1O_2...O_T$, and a model $\lambda = (A, B, \pi)$ what is the probability that the observation sequence O was produced by the model λ, namely $P(O|\lambda)$?

The answer to this question is what allows the system to do recognition. Suppose we have a number of HMMs each corresponding to a word in the recognition task. To recognize a given observation sequence we have to answer to the previous question for each model. We choose as the recognition's result the model which produces the biggest probability. A related question is:

Given the observation sequence $O = O_1O_2...O_T$, and a model $\lambda = (A, B, \pi)$, what is the state sequence $Q = q_1q_2...q_T$ which optimally explains the observation sequence?

The answer to this question provides insights on the underling physical process that is modelled. The analyst can decide to change the number of states and the set of possible observations of the model. The last meaningful question from the point of view of real applications is:

How do we build the model $\lambda = (A, B, \pi)$?

In other words, the last question can be reformulated as: After we have decided on the number of states and the number of possible observations, and we have an initial guess about the probability distributions π, A and B how do we adjust these probability distributions such that to maximize the probability $P(O|\lambda)$? The process that tries to answer this question is called "training", in other words, the best model parameters are learned based on a set of data for which we already know the classification. This is called *supervised training.* In the case of lip reading and speech recognition the most important questions are therefore question 3 and question 1. We first train the models based on the data corpus available, therefore using the question 3, and afterwards we test the resulting system for a number of test utterances, therefore using the question 1. The rest of this section will give the most popular formal mathematical solutions to the above questions.

3.1.1 The Forward-Backward Algorithm

From the theoretical point of view, computing the probability that a given observation sequence $O = O_1O_2...O_T$ was produced by a given model $\lambda = (A, B, \pi)$ is relatively straightforward because it only requires the definition. However, this solution is not feasible from the implementation point of view. The Forward-Backward algorithm is a procedure that overcomes the computational complexity.

We start by realising that the observation sequence O can be produced by taking any route Q of length T in the model, of course with a different probability. The probability that the sequence O is produced by the model λ, while traversing the state sequence $Q = q_1q_2...q_T$ is denoted by $P(O, Q|\lambda)$. From this quantity we can compute the probability of appearance of the observation sequence O by integrating out Q, (i.e. summation over all possible state sequences of length T). We have, therefore:

$$P(O|\lambda) = \sum_{\text{all } Q} P(O, Q|\lambda), \tag{3.10}$$

which can be rewritten using the chain rule as:

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda)P(Q|\lambda). \tag{3.11}$$

All we need to do now is to compute the two probabilities in the summation. First let fix $Q = q_1q_2...q_T$ a state sequence. The probability that the given observation sequence was produced by the model λ while following the state sequence Q is given by:

$$\begin{aligned} P(O|Q, \lambda) &= P(O_1O_2...O_T|q_1q_2...q_T, \lambda) \\ &= \prod_{t=1}^{T} P(O_t|q_t, \lambda) = \prod_{t=1}^{T} b_{q_t}(O_t). \end{aligned} \tag{3.12}$$

In the above derivation, statistical independence of the observations was assumed. This assumption is very important from the computation point of view and even though is not always true in most of the real applications the power of the HMMs paradigm is not impeded. The probability that such a state sequence can be generated is computed by:

$$\begin{aligned} P(Q|\lambda) &= P(q_1q_2...q_T|\lambda) \\ &= P(q_1|\lambda) \prod_{t=1}^{T-1} P(q_{t+1}|q_t, \lambda) \\ &= \pi_{q_1} \prod_{t=1}^{T-1} a_{q_tq_{t+1}}. \end{aligned} \tag{3.13}$$

Therefore, based on the equations 3.11, 3.12 and 3.13 the probability of appearance of the observation O given the model λ is computed by:

$$\begin{aligned} P(O|\lambda) &= \sum_{q_1,q_2,\ldots,q_T} \left(\prod_{t=1}^{T} b_{q_t}(O_t)\right)\left(\pi_{q_1}\prod_{t=1}^{T-1} a_{q_tq_{t+1}}\right) \\ &= \sum_{q_1,q_2,\ldots,q_T} \left(\pi_{q_1} b_{q_1}(O_1)\prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(O_t)\right). \end{aligned} \tag{3.14}$$

As we already mentioned in the beginning, this brute computation solution, even though it is very easy to understand and follow, has an extremely computational intensive derivation. It can be computed that this solution needs on the order of $2TN^T$ operations. The Forward-Backward algorithm starts by building for each partial sequence of observations $O_1O_2...O_t$ the variable $\alpha_t(j)$ that represents the probability to see the length t partial observations sequence and arrive in state S_j at time t

$$\alpha_t(j) = P(O_1O_2...O_t, q_t = S_j|\lambda). \tag{3.15}$$

The observation here is that by summation over all j of $\alpha_T(j)$ we arrive to the solution of the first question:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j). \tag{3.16}$$

The computation of the variable $\alpha_t(j)$ can be computed iteratively for all t and all states j starting with

$$\begin{aligned} \alpha_1(j) &= P(O_1, q_1 = S_j|\lambda) \\ &= P(O_1|q_1 = S_j, \lambda)P(q_1 = S_j|\lambda) = \pi_j b_j(O_1). \end{aligned} \tag{3.17}$$

The induction step is

$$\begin{aligned} \alpha_{t+1}(j) &= P(O_1O_2...O_{t+1}, q_{t+1} = S_j|\lambda) \\ &= \sum_{q_1,q_2,\ldots,q_{t+1}=j} \left(\pi_{q_1}(O_1)\prod_{\tau=2}^{t+1} a_{q_{\tau-1}q_\tau} b_{q_\tau}(O_t)\right) \\ &= \sum_{q_1,q_2,\ldots,q_{t+1}=j} p_{q_1} b_{q_1}(O_1) a_{q_1q_2} b_{q_2}(O_2)...a_{q_{t-1}q_t} b_{q_t}(O_t) a_{q_tj} b_j(O_{t+1}) \\ &= \sum_{i=1}^{N}\left(\sum_{q_1,q_2,\ldots,q_t=i} p_{q_1} b_{q_1}(O_1)...a_{q_{t-1}i} b_i(O_t)\right) a_{ij} b_j(t+1) \\ &= \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right). \end{aligned} \tag{3.18}$$

The equation 3.18 holds for any $1 \leq t \leq T-1$ and any $1 \leq j \leq N$. The computation of $P(O|\lambda)$ through the means of the variable $\alpha_t(j)$ is known as the *forward pass* and needs only on the order of N^2T operations. As the name suggests, there is a second part of the algorithm called the *backward pass.* This step is actually useful to answer question 3. We include the backward pass here to complete the forward-backward algorithm, but the completion of the answer to question 3 will be given in its corresponding section. In a similar way the backward pass computes a backward variable $\beta_t(i)$ defined as the probability of the partial observation sequence starting at time $t+1$, given that the system is in state S_i at time t and given the models λ

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T|q_t = S_i, \lambda). \tag{3.19}$$

The computation of $\beta_t(i)$ is done again by induction, namely:

$$\beta_T(i) = 1,\ 1 \leq i \leq N,\ and \tag{3.20}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(O_{t+1}\beta_{t+1}(j), \tag{3.21}$$

for $t = T-1, T-2, ..., 1, 1 \leq i \leq N$.

The computation of $\beta_t(i)$ requires N^2T operations. The continuation of the solution for question 3, starting from the backward step, is given later.

3.1.2 The Viterbi Algorithm

The first question has an exact solution. This is not the case for the second question because there is no established definition of the goal: the state sequence that *optimally explains the observation sequence.* The most widely used definition of the optimality is the single best state sequence that maximizes $P(Q|O, \lambda)$. The algorithm that solves this optimization problem is called the Viterbi algorithm. The Viterbi algorithm was conceived by Andrew Viterbi in 1967 ([Vit67]) as an error-correction scheme for noisy digital communication links. A very good tutorial to the algorithm is given by Forney in [For73]. The Viterbi algorithm is very similar to the forward step, the difference being the use of the maximum operator in the equations 3.18 and 3.16. Let $Q = \{q_1q_2...q_T\}$ be the single best state sequence for the given observation sequence $O = \{O_1O_2...O_T\}$. Similarly to the forward pass we define

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P(O_1O_2...O_t, q_1q_2...q_t = i|\lambda). \tag{3.22}$$

The $\delta_t(i)$ is the maximum probability to produce the partial observations sequence by following a fixed state path that ends up in the state i. The difference with the definition of $\alpha_t(i)$ is that the entire state path is given and not only the final state. At the first time step we have

$$\delta_1(i) = P(q_1 = i, O_1|\lambda) = P(O_1|q_1 = i, \lambda)P(q_1 = i|\lambda) = \pi_i b_i(O_1). \tag{3.23}$$

By induction we have

$$\delta_{t+1}(j) = \max_{i=1..N} (\delta_t(i)a_{ij})\, b_j(O_{t+1}), \tag{3.24}$$

and the probability of the single most probable path is found as

$$P^*(Q^*|O,\lambda) = \max_{i=1...N} \delta_T i. \tag{3.25}$$

The actual path Q is found by recording the argument of equation 3.24. The algorithm does this by using the array $\phi_t(j)$ defined in the induction step as

$$\phi_t(j) = \operatorname*{argmax}_{1\leq i\leq N}(\delta_{t-1}(i)a_{ij}). \tag{3.26}$$

In the end, the path is recovered as follows:

$$q_T^* = \operatorname*{argmax}_{1\leq i\leq N}(\delta_T(i)) \tag{3.27}$$

$$q_t^* = \phi_{t+1}(q_{t+1}^*), t = T-1, T-2,\ ...,\ 1. \tag{3.28}$$

3.1.3 The Baum-Welch Algorithm

Training the model parameters A, B and π is the most important problem, since this is what makes the HMM paradigm usable for recognition. Starting from an initial guess, we need to adapt the model parameters such that they accurately match the given real process we want to model. If this were not possible then the whole problem would be almost insolvable. There is no known analytical solution to this problem. The Baum-Welch algorithm is an iterative procedure that adjusts λ such that $P(O|\lambda)$ is locally maximized. The algorithm starts from the forward-backward variables, namely $\alpha_t(i)$ and $\beta_t(i)$. Next, it defines the probability $\xi_t(i,j)$ of being in state S_i at time t and in state S_j at time $t+1$ given the observation O and the model λ and the probability $\gamma_t(i)$ of being in state S_i at time t given the observation O and the model λ. The two probabilities are defined as follows:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j|O,\lambda),\ \textit{and} \tag{3.29}$$

$$\gamma_t(i) = P(q_t = S_i|O,\lambda). \tag{3.30}$$

Based on the forward and backward probabilities we have

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)},\ \textit{and} \tag{3.31}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}. \tag{3.32}$$

The summation over time of the two quantities above have the following interpretations which are extremely important to understand the solution of the Baum-Welch algorithm:

$$\sum_{t=1}^{T} \gamma_t(i) = \text{the expected number of times in state } S_i, \tag{3.33}$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{the expected number of transitions from } S_i, \tag{3.34}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{the expected number of transitions from } S_i \text{ to } S_j. \tag{3.35}$$

Finally, using the above interpretations and the probabilities defined in 3.31 and 3.32 the re-estimated parameters of the model λ, as defined by the Baum-Welch approach are:

$$\overline{\pi_i} = \gamma_1(i), \tag{3.36}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \; \textit{and} \tag{3.37}$$

$$\overline{b_j(k)} = \frac{\sum_{t=1, O_t=V_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{3.38}$$

The re-estimated model's parameters based on the Baum-Welch approach are proved to converge to a local optimum; therefore, the initial guess is very important.

3.2 Principal Component Analysis

The *Principal Component Analysis*(PCA) is one of the most used tools in data mining and patterns recognition. It is a statistical technique for finding patterns in high dimension data. It actually finds the directions (i.e. principal components) that explain most of the variance in the data. In the lip reading domain PCA was extensively used for data parametrization built around the notion of eigenfaces, for dimension reduction of the feature space but also for tracking of key points as in the AAM technique. The PCA projection also produces a de-correlation of the feature dimensions.

The PCA was introduced by Karl Pearson in 1901 in [Pea01]. In mathematical terms it is defined as an orthogonal linear transformation that projects the data to a coordinate system such that the greatest variance lies on the first coordinate, called the first principal component, the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for given data in least square terms. The PCA involves the calculation of the eigenvalue decomposition of the mean centred data covariance matrix. The main tools used for PCA are therefore: sample mean, sample variance, sample covariance matrix, the eigenvectors and eigenvalues of the covariance matrix. PCA is very simple to implement but a very powerful tool which we used in many places in our research.

Let $X_1, X_2, ...X_N$ be the data needed to be processed. Each vector X_n has M components and is considered as a sample from a multi-variate distribution. In our case M could represent the number of pixels in the face image, or the number of feature vectors. N is the number of images that are being processed. Create the $(M \times N)$ matrix X, which has as columns the vectors X_n. The first necessary step is to centre the data by subtracting the sample mean from each dimension of the data:

$$\overline{X_m} = \frac{1}{N}\sum_{n=1}^{N} X_{mn}, \text{ for all } m = 1, ..., M \tag{3.39}$$

$$Y = Y_n, 1 \le n \le N, \tag{3.40}$$

where the columns are computed as $Y_n = X_n - \overline{X}$.

The covariance matrix of the centred data is computed as follows

$$C = \frac{1}{N}\sum YY^*, \text{ where “*” represents the conjugate transpose of Y,} \tag{3.41}$$

based on the sample (co)variance estimation given by

$$Var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})', \tag{3.42}$$

$$Cov(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})'. \tag{3.43}$$

The next step is to compute the $(M \times M)$ matrix V of eigenvectors such that

$$V^{-1}CV = D, \tag{3.44}$$

where D is a diagonal matrix which has on the diagonal the corresponding eigenvalues λ. Therefore, each column V_m in the matrix V is an eigenvector of the covariance matrix and corresponds to the eigenvalue $\lambda_m = D_{mm}$ in the matrix D. The value of λ_m gives the contribution to the variance of the data captured by the m^{th} eigenvector (i.e. m^{th} principal component). By sorting the eigenvectors according to their contribution and computing the cumulative contribution of the principal components it is possible to decide the sufficient number of components. Usually, only the components that cover at least 90% to 95% of the variance are considered. If

$g(m) = \sum_{i=1}^{m} D_{ii}$ is the cumulative contribution until the m^{th} component then only the first L directions such that $g(L) = 95\% g(M)$ are chosen. The data matrix is transformed into the new space through the $(M \times L)$ matrix W having as columns the first L eigenvectors

$$Z = W^{*}Y. \tag{3.45}$$

Since $L \leq M$, the result is also a compression of the data. However, a lossy compression since it is not possible to get back the original data. The results on a small $2D$ artificial data set are shown below. Let $N = 20$, $M = 2$ and the data given in the Table 3.1.

Table 3.1: *Sample data for PCA. In the left is the original data and in the right is the centred data.*

	X_1	X_2		Y_1	Y_2
	9.8162	25.1921		-0.1617	0.1149
	10.2384	24.7629		0.2605	-0.3143
	10.2903	24.8370		0.3124	-0.2402
	9.9003	24.9688		-0.0776	-0.1085
	9.8999	25.3217		-0.0780	0.2444
	9.7517	25.0426		-0.2262	-0.0346
	9.5919	25.4296		-0.3860	0.3523
	10.1022	25.0995		0.1243	0.0223
Data =	10.3324	24.8482	Centred data =	0.3545	-0.2290
	9.7935	25.2723		-0.1844	0.1950
	9.9655	24.9156		-0.0124	-0.1616
	9.5455	25.3940		-0.4324	0.3167
	9.5857	25.3329		-0.3922	0.2557
	10.0131	24.8942		0.0352	-0.1831
	10.2154	25.0081		0.2375	-0.0691
	9.5354	25.4821		-0.4425	0.4049
	10.4796	24.6214		0.5017	-0.4559
	9.9974	25.1901		0.0195	0.1128
	10.2007	25.0424		0.2228	-0.0349
	10.3031	24.8895		0.3252	-0.1877

The covariance matrix computed based on the centred data is

$$cov = \begin{pmatrix} 0.0832 & -0.0611 \\ -0.0611 & 0.0583 \end{pmatrix},$$

and the eigenvectors and the eigenvalues of this covariance matrix are:

$$eigenvalues = \begin{pmatrix} 0.1331 \\ 0.0084 \end{pmatrix},$$

and

$$eigenvectors = \begin{pmatrix} 0.7744 & 0.6327 \\ \text{-}0.6327 & 0.7744 \end{pmatrix}.$$

In Figure 3.2 the principal components superimposed on the centred data points are shown. In Figure 3.3 the projected data on the space determined by the two principal components is shown. The two figures show on both spaces the data points modified by one unit in the direction of the less important principal component. The adjustment is done in the projected space. As expected, this adjustment shows up in the centred data space as a translation orthogonal on the first principal component. This thinking is used in the AAM method during the iterative searching scheme. The step in the projected space is chosen as a factor of the variance covered by the corresponding principal component.

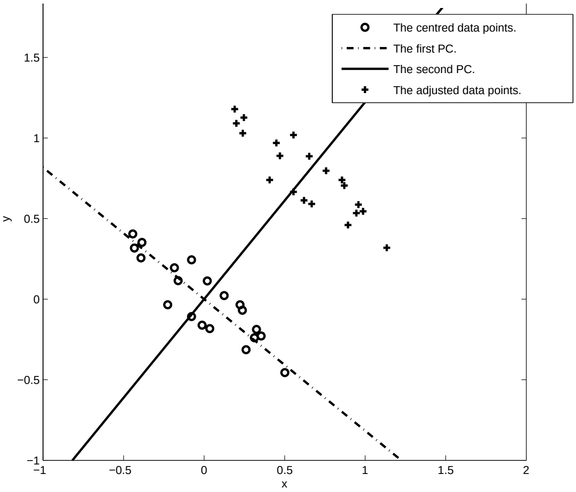


Figure 3.2: *PCA example data: data is centred around the mean. The computed PCs are shown superimposed on the centred data. The figure shows a second set of points obtained from the original points by translation with one unit in the PCA space.*

The use of eigenfaces for the characterization of human faces for pattern recognition applications was first used by Kirby and Sirovich in [Sir87; Kir90]. Turk and Pentland obtained in 1991 the first promising result for the problem of face recognition from static images, using the PCA approach ([Tur91]). In their paper Turk and Pentland derived an alternative computation which makes the PCA suitable for processing images. The problem comes from the high dimensionality of the feature space when images are considered. For instance, for an image of size (100 × 100) the PCA problem is moving in a 10000 dimensional space and the matrix data will be ($M \times 10000$). From the practical point of view computing the eigenvectors of the (10000 × 10000) covariance matrix is not feasible. The covariance matrix as computed in 3.41 can be written as

$$C = YY^T, \tag{3.46}$$

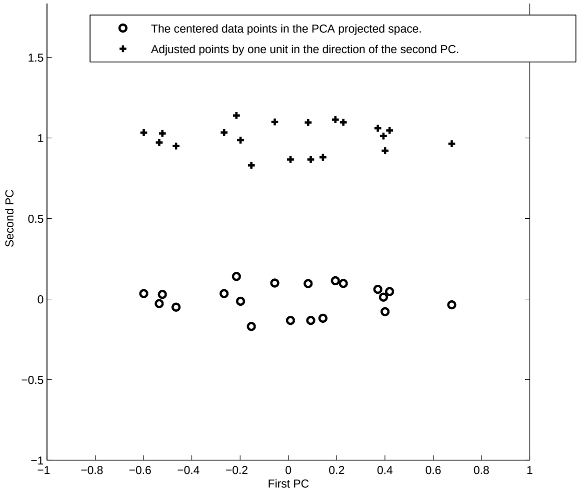


Figure 3.3: *PCA example data: data is projected in the principal components space. The figure shows a second set of points obtained from the original points by translation with one unit in the positive direction of the y axis.*

where Y is the matrix $Y = (Y_1Y_2...Y_N)$. The eigenvectors of the matrix C are defined as $V_m \neq 0$ such that

$$CV_m = YY^TV_m = \lambda_m V_m. \tag{3.47}$$

Turk and Pentland observed that if we instead compute the eigenvectors U_m of Y^TY which is an $M \times M$ matrix we have

$$Y^TYU_m = \lambda_m U_m, \tag{3.48}$$

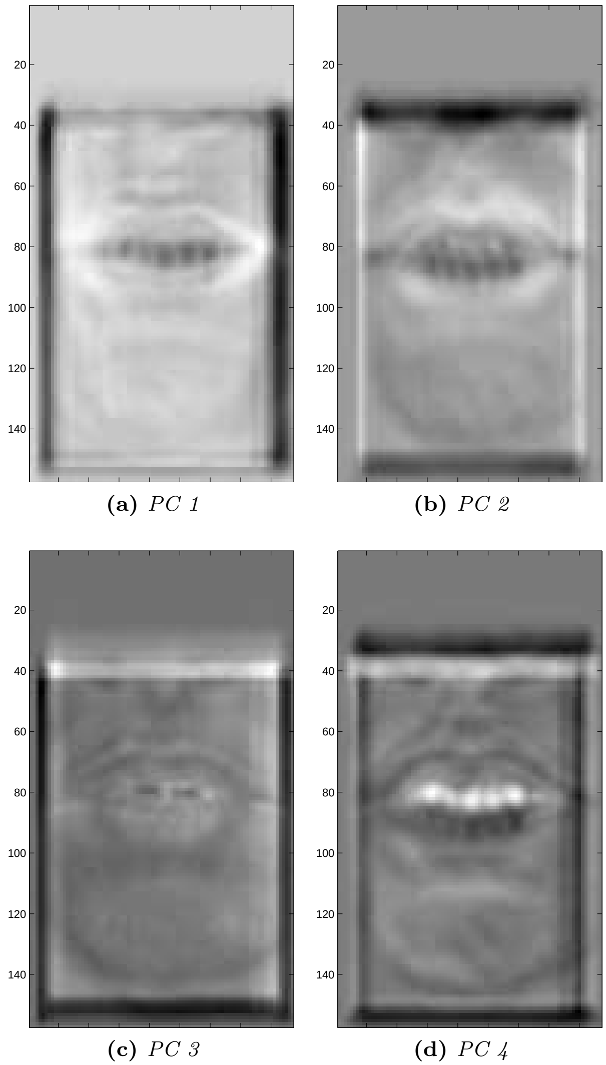
and if we multiply the above equation with Y to the left we obtain

$$\left(YY^T\right)(YU_m) = \lambda_m (YU_m). \tag{3.49}$$

This means that if U_m is an eigenvector of the matrix Y^TY corresponding to the eigenvalue λ_m then the vector YU_m is an eigenvector of the matrix YY^T corresponding to the same eigenvalue.

Figure 3.4 shows the first eigenfaces computed for a small subset (e.g. 1156 image files) of the data corpus used in our work. It is interesting to find that only a small number of components are needed to capture a large portion of the variance in the data. The first four eigenfaces shown in the image already account for more the 70% of the variance.

In Figure 3.5 the results obtained in this case for an experiment similar to the one shown in Figures 3.3 and 3.2 are presented.



(a) *PC 1*

(b) *PC 2*

(c) *PC 3*

(d) *PC 4*

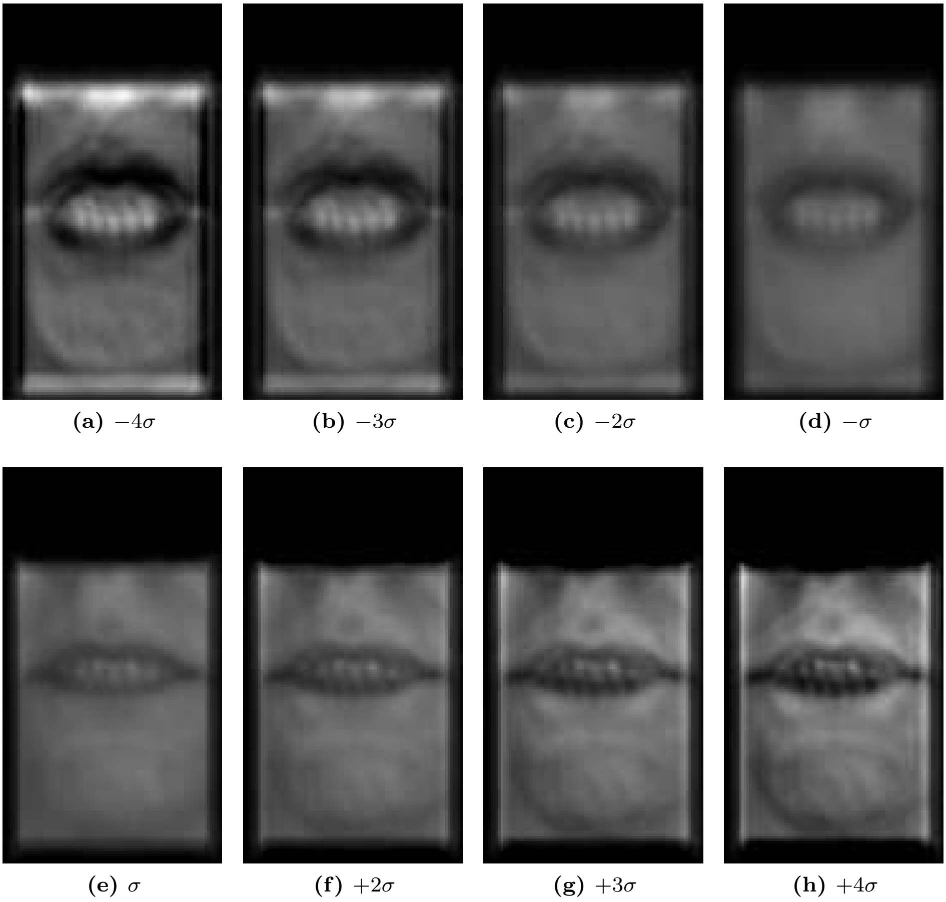
Figure 3.4: *PCA example for image processing. The images represent the first four principal components that account for 70% of the variance in the data.*

3.3 Optical Flow Analysis

The Optical Flow concerns the apparent motion of objects within a visual representation. A common definition of the Optical Flow is:

The velocity field which warps one video frame in a subsequent one.

In [Hor81] the optical flow is defined as:



(a) -4σ **(b)** -3σ **(c)** -2σ **(d)** $-\sigma$

(e) σ **(f)** $+2\sigma$ **(g)** $+3\sigma$ **(h)** $+4\sigma$

Figure 3.5: *The images obtained by perturbing, in the PCA space on the second principal direction by* -4σ, -3σ, -2σ, $-\sigma$, σ, 2σ, 3σ, 4σ, *respectively, the mean image.* σ *represents the standard deviation computed for the data corpus for the second principal direction.*

The distribution of apparent velocities of movement of brightness patterns in an image.

The word *apparent* is used in the definition because the optical flow does not consider the movement of the objects in the real $3D$ space but the motion in the image space. Therefore, sometimes the optical flow does not have the same orientation as the true motion field. A well known example is the *rotating barber's pole illusion.* The problem of finding the optical flow in an image falls in a broader class of problems called *image registration problem.* Data registration in general deals with spatial and temporal alignment of objects within imagery or spatial data sets. Image registration can occur at pixel level (i.e. any pixel in an image can be matched with known accuracy with a pixel or pixels in another image) or at object level (i.e. it relates objects rather than pixels). The domain where the image registration problem is one of the key challenges is medical imaging. In medical imaging the problem of registration arises whenever images acquired from different subjects, at different times, or from different scanners need to be combined for analysis or visualization. In [Luc81] the problem of finding the optical flow seen as an image registration problem is defined as follows:

We consider that the pixels values in the two images are given by the functions $F(X)$ and $G(X)$ (in 2D space $X = (x, y)$). Our goal is to determine the dissimilarity vector h which minimizes some measure of the difference between $F(X + h)$ and $G(X)$, for X in some region of interest $\Re$.

There are many different methods for determining the optical flow depending on the mathematical interpretation of the problem: block based differences, inverse of normalized cross-power spectrum, discrete optimization methods, differential methods, etc. The most successful approaches were obtained by Lucas and Kanade ([Luc81]) and Horn and Schunck ([Hor81]) both methods being based on partial derivatives of the image signal. The first approach defines a set of image windows and an affine model of the flow field. This algorithm assumes that the images are roughly aligned and that the optical flow is constant in a small neighbourhood. Then it uses a type of Newton-Raphson iteration taking the gradient of the error and assuming that the analyzed function is almost linear and it moves in the direction of this gradient. The latter algorithm assumes that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. The algorithm minimizes the square of the magnitude of the gradient of the optical flow velocity and the measure of non-smoothness of the optical flow. In [Bru05] the authors explore the possibility of combining the two approaches used in Lucas-Kanade and Horn-Schunck methods, namely local constraints methods and global constraints methods, in order to build a hybrid method that can provide the corroborated strengths of both paradigms.

Other known algorithms are developed by Uras et al. [Ura88], Nagel [Nag87], Anandan [Ana89], Singh[Sin91], Heeger [Hee87], Waxman et al. [Wax88], Brox et al. [Bro04] and Fleet and Jepson [Fle90]. In [Bar94] and [Gal98] a number of nine, respectively eight, different techniques for detection of optical flow were investigated. The performance of these methods was compared on synthetic scenes. The difficulty of comparing different optical flow techniques comes from the fact that it is hard to produce ground-truth motion fields. In [Gal98] this problem was overcome with a modified ray tracer that allowed the generation of ground-truth flow maps. The latter study reports that a modified version of Lucas-Kanade algorithm produced the best results, however, the algorithm does not produce dense flow maps. On the second place was ranked Proesmans algorithm which produced very dense flow maps but with less quality. Baker et al.([Bak07]) build a corpus for evaluation of the various approaches and implementations of the solutions to the optical flow problem. In their experiments the authors concluded that the pyramidal Lucas-Kanade algorithm performs better than the original algorithm.

For our research, we had two measures for the performance of the optical flow detection method, namely the accuracy with which the flow is detected including the resolution of the flow field and the computational complexity of the algorithm expressed as the duration for obtaining the flow. For our research we used two variants, a pyramidal implementation of the Lucas-Kanade algorithm and an implementation of the Horn-Schunck algorithm.

3.3.1 Differential Approach Overview

The problem of determining the optical flow as a differential problem is introduced below. Consider the function $I(x, y, t)$ which gives the image intensity at location (x, y) at time t. Every optical flow detection method has as goal to compute the motion of every pixel in the image from time t to time $t + \delta t$. If we denote the new position of the pixel (x, y) from time t with $(x + \delta x, y + \delta y)$ at time $t + \delta t$ we get the following constraint equation:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{3.50}$$

Assuming that the movement inside the image is small enough, the image constraint equation can be rewritten in terms of Taylor series as follows:

$$\begin{aligned} I(x, y, t) &= I(x + \delta x, y + \delta y, t + \delta t) \\ &= I(x, y, t) + \tfrac{\partial I}{\partial x}\delta x + \tfrac{\partial I}{\partial y}\delta y + \tfrac{\partial I}{\partial t}\delta t + \Re, \end{aligned} \tag{3.51}$$

where $\Re$ means higher order terms, which are small enough to be ignored. Using the initial image constraint and ignoring $\Re$ we get the following equation:

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0, \tag{3.52}$$

which can be rewritten as:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y = -\frac{\partial I}{\partial t}, \tag{3.53}$$

where V_x and V_y are the components of the optical flow. Denoting the partial derivatives of $I(x, y, t)$ with respect to spatial coordinates x and y and time t with I_x, I_y and I_t respectively the new constraint equation reads:

$$I_x V_x + I_y V_y = -I_t. \tag{3.54}$$

Hence the problem of detecting optical flow is equivalent to solving the system (3.54). However, this system has only one equation but two unknowns making it under-determined. To be able to solve this system some assumptions need to be taken. Based on these assumptions new equations can be introduced. The resulting flow obtained will carry the marks of these assumptions.

3.3.2 Lucas-Kanade Algorithm

Lucas and Kanade's algorithm starts with the assumption that optical flow is constant in a small neighbourhood of the point (x, y). Assuming that the flow (V_x, V_y) is constant in a small rectangular region of size (n, n) with $n > 1$ (usually $n = 5$ gives sufficient good results), that is centred at point (x, y) and numbering the pixels, we get the following system:

$$\begin{cases} I_{x_1}V_x + I_{y_1}V_y = I_{t_1} \\ \cdots\cdots \\ I_{x_n}V_x + I_{y_n}V_y = I_{t_n} \end{cases}, \tag{3.55}$$

which is an over-determined system. Written in matrix form it reads:

$$\begin{bmatrix} I_{x_1} & I_{y_1} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -I_{t_1} \\ \vdots \\ -I_{t_n} \end{bmatrix}. \tag{3.56}$$

A weighted least squares fit solution of the above system is:

$$V = [A^T W A]^{-1}(-A^T W b), \tag{3.57}$$

where $A = \begin{bmatrix} I_{x_1} & I_{y_1} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix}$, $V = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$, $b = \begin{bmatrix} I_{t_1} \\ \vdots \\ I_{t_n} \end{bmatrix}$ and W is a weighting function that gives more importance to the centre pixel of the window. This means that the optical flow vector can be found only by calculating the derivatives of the image in all dimensions. The Lucas-Kanade optical flow detection algorithm is very fast because it examines only a limited number of possible matches, however, with the disadvantage that it does not yield a high density of flow vectors, (i.e. the velocity is only determined close to the boundaries of objects and inside large areas with almost constant brightness the information fades quickly). The main advantage is the algorithm robustness in the presence of noise.

3.3.3 Horn-Schunck Algorithm

The Horn-Schunck method uses a global constraint of smoothness in order to solve the aperture problem. Starting from the equation (3.54) the smoothness constraint is added as the necessity to minimize the square of the magnitude of the gradient of the optical flow, namely the following two quantities:

$$\left(\frac{\partial V_x}{\partial x}\right)^2 + \left(\frac{\partial V_x}{\partial y}\right)^2 \text{and} \left(\frac{\partial V_y}{\partial x}\right)^2 + \left(\frac{\partial V_y}{\partial y}\right)^2. \tag{3.58}$$

In order to solve the problem Horn and Schunck consider the following minimization problems:

$$\begin{cases} \underset{(V_x,V_y)}{\operatorname{argmin}}(I_xV_x + I_yV_y + I_t) \\ \underset{(V_x,V_y)}{\operatorname{argmin}}\Big((\frac{\partial V_x}{\partial x})^2 + (\frac{\partial V_x}{\partial y})^2 + (\frac{\partial V_y}{\partial x})^2 + (\frac{\partial V_y}{\partial y})^2\Big) \end{cases}, \tag{3.59}$$

with the first equation also seen as a minimization problem. The first problem asks for the minimization of the sum of errors in the equation of the rate of change of image brightness, while the second asks for the minimization of the measure of the departure from the smoothness in the velocity flow. Hence the function to be minimized reads:

$$f = \int ((\nabla I \bullet \vec{V} + I_t)^2 + \alpha(|\nabla V_x|^2 + |\nabla V_y|^2))dxdy, \qquad (3.60)$$

where $\nabla I = \begin{bmatrix} I_x \\ I_y \\ I_t \end{bmatrix}$, and $\vec{V} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$ is a regularization constant; larger values of α lead to a smoother flow. Minimizing the above function comes down to solving the corresponding Euler-Lagrange equations. An iterative solution of the minimization problem will have the following updates:

$$\begin{cases} V_x^{k+1} = \bar{V}_x^k - \frac{I_x[I_x\bar{V}_x^k + I_y\bar{V}_y^k + I_t]}{\alpha^2 + {I_x}^2 + {I_y}^2} \\ \\ V_y^{k+1} = \bar{V}_y^k - \frac{I_y[I_x\bar{V}_x^k + I_y\bar{V}_y^k + I_t]}{\alpha^2 + {I_x}^2 + {I_y}^2} \end{cases}, \qquad (3.61)$$

where $\bar{V}$ denotes region average and k denotes the iteration number.

Compared to the Lucas-Kanade method, the Horn-Schunck algorithm yields a high density of flow vectors. However, it is more sensitive to noise and has problems with spatial discontinuities in image brightness (e.g. when there is an object that is occluded by other objects.)

3.3.4 Performance Measures

The most used measure of accuracy to optical flow is the *Angular Error*(AE). The AE between two flows $V_0 = (u_0, v_0)$ and $V_1 = (u_1, v_1)$ is computed as the inverse cosine of the dot product of the normalised vectors in the space time domain.

$$\Psi_\epsilon = \arccos\left[\frac{<(V,1),\ (U,1)>}{\sqrt{1+||V||^2}\sqrt{1+||U||^2}}\right]. \qquad (3.62)$$

The error in flow endpoint defined by $\sqrt{[(u0-u1)^2 + (v0-v1)^2]}$ is also used. In all cases the mean and the standard deviations are reported in order to describe the performance on the entire image.

In the paper [Bak07] the authors compute the square root of the normalized sum of squared difference between an interpolated image $I(x,y)$ and a ground truth image $I_{GT}(x,y)$ as a measure of performance. This is defined by:

$$\left[\sum_{(x,y)} \frac{(I(x,y) - I_{GT}(x,y))^2}{||\nabla I_{GT}(x,y)||^2 + 1}\right]^{\frac{1}{2}}. \qquad (3.63)$$

Another important aspect in choosing a suitable algorithm for optical flow detection is also the performance in terms of computational complexity. For a real system there is a strong requirement to have a real time performance. Usually, a system is characterized in terms of a multiple of real time. A real time system, is a system that has a score of 1 which means that at most one time unit is necessary to process one time unit of corresponding data. In our case however, due to the large amount of data needed to process, we have chosen the implementation that needed the least amount of time to compute the optical flow while keeping the accuracy in reasonable limits.

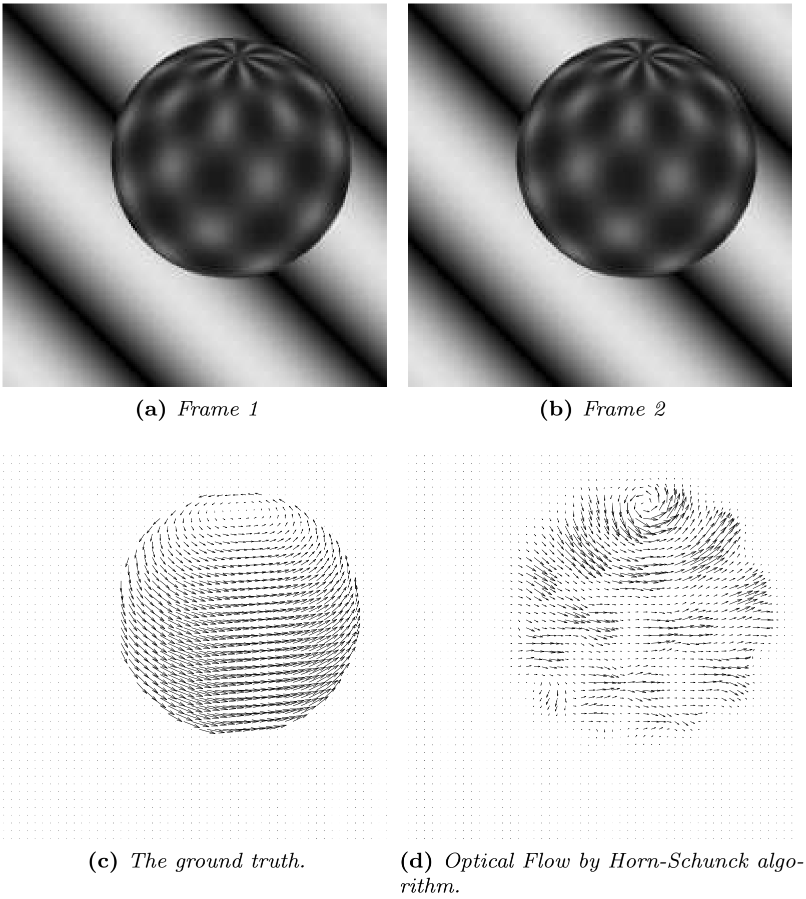
In our work we considered both a pyramidal implementation of the Lucas-Kanade algorithm and an implementation of Horn-Schunck algorithm. However, the implementation of the latter was considerably faster (e.g. approximately 5-10 seconds were needed for the Horn-Schunck algorithm while around 170-200 seconds were needed for the Lucas-Kanade implementation) with no visible loss of accuracy. We opted, therefore, for the Horn-Schunck method.

Image 3.6 shows an example of the performance of the Horn-Schunck algorithm on a succession of two synthetic images containing a rotating sphere. The ground truth computed using a ray tracer is also shown. The mean and standard error of the angular error computed as in [Bar94; Fle90], by equation 3.62, for the given example are 16.98 and 26.42, respectively. The component wise measures for the angular error as defined in [Bar94] are 14.93 and 24.08, respectively. The mean of the component wise square root differences, namely the mean of the error in the endpoint, is 0.5 pixels.

3.4 Active Appearance Models

Active Appearance Models(AAM) is a model based approach for image segmentation. It was first introduced by Edwards, Taylor and Cootes in [Edw98] for interpreting face images. It is a top-down approach because it is based on a-priori knowledge about shape and appearance of the targeted object, therefore, it can only be applied in situations when the test images are already validated to contain the targeted object. During the training phase a statistical model of the shape and appearance of the object is built starting from the variation induced by the physical phenomena that governs the object's image. This approach is closely related to the *Active Blobs Models* and *Active Contours Models* also called *"Smart" Snakes* ([Kas88]), and *Active Shape Models* ([Coo92]) all being in general a type of *Morphable Models* ([Bla99]). The main difference with all these approaches is that AAM makes use of stronger global constraints enforced through the a-priori decision about the structure of the model. The powerful generalization to unseen instances makes them extremely useful in various applications, however, the most important being the medical applications. A good overview on the AAM and similar models to medical image analysis is done in [Coo01].

The method assumes that a set of *landmark* points can always be marked on the image to describe the shape of the object. The transformation of the object should, therefore, not degenerate such that the points are overlapping or are occluded. From a set of images which have the landmarks already marked the method generates a statistical model of the shape variation, a model of the texture variation and a model of the correlations between the shape and the texture. Based on these models, the



(a) *Frame 1*

(b) *Frame 2*

(c) *The ground truth.*

(d) *Optical Flow by Horn-Schunck algorithm.*

Figure 3.6: *The optical flow as computed by the Horn-Schunck algorithm.*

method should be able to synthesize any new instance of the target object, even though it was not "seen" in the training set, as long as the variation from the mean shape of the object is in the learned dynamic range. The searching scheme consists of an iterative model refinement that adjusts the model parameters such that the error between the synthesized model image and the real image is minimized.

3.4.1 Shapes and Landmarks

As we already mentioned, the two main concepts on which the AAM is based are *shape* defined based on a set of *landmarks* and *texture*. However, the concept of shape is the defining one. The definitions of shape and landmarks used were adopted from [Dry98]. A *landmark is a point of correspondence on each object that matches between and within populations.* A good choice of the landmarks is very important in order to optimize the location of the object. Cotes suggested in [Coo00] that the landmarks should be chosen as the clear corners of the object, 'T' shape junctions between the boundaries or other easily to locate landmarks. In order to make the boundary of the object robust, he suggested choosing other landmarks equally spread between the main landmarks on the boundaries of the object. For a given image the shape of the object is, therefore, defined by a set of N points in the $2D$ space $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$. From this, the shape vector can be formed, for instance, as:

$$x = (x_1, x_2, ..., x_N, y_1, y_2, ..., y_N)^T. \tag{3.64}$$

The *shape* is defined as *all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.* This definition is very important since it implies that all shapes should be aligned into a common co-ordinate system. This is usually achieved by "Procrustes analysis" ([Goo91], [Boo96], [Dry98]). The shapes are, therefore, projected into a shape space. The shapes are translated, rotated and scaled such that the sum of distances ($\sum(x^i - \overline{x})$ of each shape to the mean is minimized.

3.4.2 Learning the Statistical Model of the Variance

The AAM are linear in both shape and appearance but are nonlinear in terms of pixels intensities. Therefore, the AAM allows only for linear shape and appearance variation. By the nature of the problem there are three situations that are considered: a model for shape, a model for texture and a combined model for both shape and texture. The first two models are also called independent models, because they assume that the two spaces are independent. The combined model is actually introduced to take care of the correlations between the shape and the appearance parameters.

Shape Model

As above, let the shape of the model to be specified by a set of points that make up the mesh:

$$x = (x_1, x_2, ..., x_N, y_1, y_2, ..., y_N)^T. \tag{3.65}$$

Since there can only be a linear variation of the shape, the shape x can be written starting from a base shape x^0 as follows:

$$x = x^0 + \sum_{i=1}^{N} b_i x^i. \tag{3.66}$$

In order to write the above equation in a canonical form, the vectors x^i are taken such that they are orthonormal (i.e. orthogonal of length 1). The coefficients $b = b_{i_i}$ are the parameters of the shape model.

A standard approach to obtain the above parametrization is to use Principal Component Analysis. Therefore, in the above equation x^0 is taken as the mean shape $\overline{x}$, and x^i are the first n eigenvectors, $P = x^1, x^2, ...x^n$ that account for a given percentage of the data variance. The shape model can be re-written as

$$x = \overline{x} + P_s b_s, \tag{3.67}$$

where the subscript s is added to make the difference with the appearance case. The parameters b_s can be obtained as follows:

$$b_s = P_s^T (x - \overline{x}). \tag{3.68}$$

The same approach will be used in the two other cases as well, hence the PCA ends up to be applied three times.

Appearance Model

The appearance model deals with the pixels $px = (px_x, px_y)^T$ that lie inside the mean shape. The appearance of x can be written in general as

$$A(px) = A_0(px) + \sum_{i=1}^{m} \lambda_i A_i(px), \text{ for any } px \in \text{ the mean shape } \overline{x}. \tag{3.69}$$

In terms of PCA, the above equation can be written as

$$g = \overline{g} + P_g b_g. \tag{3.70}$$

Combined Model

The combined model aggregates together both shape and texture spaces. Therefore, the combined AAM uses a single set of model parameters $c = (c_a, c_2, ..., c_l)^T$ to characterize both shape

$$x = x_0 + \sum_{i=1}^{l} c_i x^i, \tag{3.71}$$

and appearance

$$A(px) = A_0(px) + \sum_{i=1}^{l} c_i A_i(px). \tag{3.72}$$

This approach, therefore, is more general and in its general definition does not make a distinction between the shape and appearance parameters. However, rearranging the coefficients c_i such that

$$c = (p_1, p_2, ..., p_n, \lambda_1, \lambda_2, ..., \lambda_m)^T,$$

and choosing x^i and A_i appropriately, the independent models are obtained.

In [Coo98] the authors directly address the problem of the correlation that might exist between the shape model parameters and the appearance model parameters.

Therefore, a third PCA is actually performed on the parameter vectors that concatenate the shape parameters and the appearance parameters employing the de-correlation property of PCA. The shape parameters are appropriately weighted to account for the difference between the two scales in the two spaces. The combined parameter vector reads:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T(x - \overline{x}) \\ P_s^T(g - \overline{g}) \end{pmatrix}. \tag{3.73}$$

Applying PCA on the parameters b, a new set of de-correlated parameters c is produced such that

$$b = P_c c = \begin{pmatrix} P_{cs} \\ P_{cg} \end{pmatrix} c, \tag{3.74}$$

where P_c is the PCA projection matrix, since b has a zero mean. The shape and appearance parameters are then linearly re-parameterised in terms of the new eigenvectors as follows:

$$\begin{aligned} b_s &= W_s^{-1} P_{cs} c \\ b_g &= P_{cg} c. \end{aligned} \tag{3.75}$$

3.4.3 AAM Fitting

Fitting an AAM to an image is a nonlinear optimisation problem. Usually, an iterative method is used to solve for incremental additive updates to the parameters. The AAM assumes that there is a linear relation between the updates of the model parameters and the error between the synthesised image and the real image. Given the current estimates of the parameters a new model image is estimated and an error measure can be computed between the synthesized image and the real image. The searching scheme's goal is to iteratively update the parameters so that to minimize this error. The distance between the two images is given by:

$$r(b) = g_{im} - g_m, \tag{3.76}$$

where b are the current estimated parameters, g_{im} is the real image, and g_m is the generated image. The error is usually defined as the sum of squared elements of r, $E(b) = r^T r$. Therefore, fitting the AAM to a given image is performed in two steps. The first step consists of generating a new model instance based on the current parameter estimation, and the second step consists of updating the model parameters.

Generating a New Model Instance

Starting from the current model parameters c the shape and the appearance can be computed using equations 3.67 and 3.70 where the shape and model parameters are expressed using the combined parameter vector shown in 3.75.

$$\begin{aligned} x &= \overline{x} + P_s W_s^{-1} P_{cs} c \\ g &= \overline{g} + P_g P_{cg} c. \end{aligned} \tag{3.77}$$

Given the new shape x, the model image instance is obtained by using a transform S, that adjusts the location of each model point. The transform S consists of a scaling by factor s, an in-plane rotation θ and a translation by (t_x, t_y). The transformation reads:

$$S\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{pmatrix}\begin{pmatrix} x_i \\ y_i \end{pmatrix}. \tag{3.78}$$

The final texture is obtained from g by applying the transform $T = \alpha g + \beta(1, 1, ..., 1)^T$.

Iterative Model Refinement

The searching algorithm used was introduced in [Coo01]. Starting from the difference measure defined in equation 3.76, the method searches the appropriate step δp such as to minimize $|r(p + \delta p)|^2$. An approximation of the last quantity can be obtained through Taylor expansion

$$r(p + \delta p) = r(p) + \frac{\partial r}{\partial p}\delta p, \tag{3.79}$$

where $\left(\frac{\partial r}{\partial p}\right)_{ij} = \frac{dr_i}{dp_j}$. The RMS solution stating from equation 3.79 is given by:

$$\delta p = -Rr(p), \text{ where } R = \left(\frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p}\right)^{-1} \frac{\partial r}{\partial p}^T. \tag{3.80}$$

The authors of [Coo01] suggest that the matrix $\frac{\partial r}{\partial p}$ is computed only once from the training set. Starting from equation 3.80 and given the current estimate of the model parameters c, the pose t, the texture transformation u, and the current image sample, g_{im}, Cootes and Taylor suggest the following iterative process for finding the optimal update for the model parameters:

1. Wrap the texture sample into the texture model frame using $g_s = T_u^{-1}(g_{im})$.
2. Compute the error vector, $r = g_s - g_m$, and the current error, $E = |r|^2$.
3. Compute the predicted additive updates, $\delta p = -Rr(p)$.
4. Update the model parameters $p \rightarrow p + k\delta p$, where initially $k = 1$.
5. Generate a new model instance, X' and a new model frame texture g'_m.
6. Sample the image at the new points to obtain g'_{im}.
7. Compute the new error vector, $r' = T_{u'}^{-1}(g'_{im}) - g'_m$.

8. Repeat the steps 4 - 7 with $k = 0.5, k = 0.25$, etc. until $|r'|^2 < E$.

The above procedure is repeated until no improvements are made to the error r. To speed up the convergence, a multi-resolution implementation is usually used, which means that the iterative process starts in the coarsest resolution and iterate a number of steps (usually 5 steps), before projecting to a higher resolution, and continuing.

Chapter 4

Data Acquisition

The data corpus used has a great influence on the success of the scientific endeavour. However, the effort and resources necessary to build a good data corpus in the audio visual speech recognition domain are a very large inhibitor factor. As we show in Section 2.3, the number of available data corpora is very small and the available data corpora are failing in a number of aspects: language, respondents, speaking style, sampling rate. Therefore, we argue that the researchers should follow some general guidelines when building a corpus that guarantees that the resulting datasets have common properties. This will give the opportunity to compare the results of different approaches of different research groups even without sharing the same data corpus. As concluded in Section 2.3, for our research we decided to build our own corpus that satisfies the requirements we considered important for lip reading research.

This chapter describes the data corpus we developed for our work *New Delft University of Technology Audio Visual Speech Corpus* [Chi08a; Chi09a]. At first we describe the design of the recording setup and the recording sessions, and, thereafter, we include statistics about the corpus in all its important aspects.

4.1 Data Corpus Requirements

Before starting any recordings one should decide on the characteristics of the resulting corpus. The following list contains all the requirements we had for our corpus. The argumentation for the decisions we made is found in Chapter 2, more thoroughly in Section 2.3. We decided that the resulting corpus will have the following characteristics:

1. The corpus is intended for the Dutch language. This means that only Dutch native speakers will be asked to participate in the recording sessions.

2. The corpus is made for user independent lip reading, which means that there will be more users recorded and large user variability will be required.

3. The Region Of Interest (ROI) will be fixed to the lower half of the speaker's face.

4. The recordings will contain synchronized frontal and side view.

5. The resolution of the images will be fixed at $\frac{1}{2}$PAL which is actually 384×288 pixels.

6. The video recordings will be made at 100 Hz.

7. The audio recordings will be made using a 48 kHz sample rate and 16 bit sample size.

8. The recordings will be made in a controlled environment, which means controlled illumination and controlled acoustic environment.

9. The language data will deal with several recognition tasks, namely, connected digits, connected strings, random words, context aware fixed grammar sentences and continuous speech random sentences.

10. The corpus will contain different speaking styles, namely, whispering, high speech rate and low speech rate.

4.2 Recording Setup

In the case of video data recording there are a larger number of important factors that control the success of the resulting data corpus. Hence, not only the environment but also the equipment used for recording and other settings actively influence the final result. Figure 4.1 shows the physical setup during the recordings. This section provides details on every component of this setup: video devices, audio devices, side view setup, illumination and acoustics and prompter software. The entire system was integrated by a Windows XP SP3 machine with 4GB of RAM and a dual core Intel CPU. The most important feature of the computing system was a 596GB RAID0 software based disk configuration with 4 disks. This was extremely important for the recordings since it was the only way we could achieve the necessary writing rate to cope with the large bandwidth data streams. The first solution we thought of was to use a RAM drive, but this configuration constrained us to a maximum of 30-40 seconds of continuous recording. We had to stop the recording every time the memory was full and wait for the transfer of data to the disk to take place. This operation turned out to be very lengthy sometimes and made the recording sessions very difficult. Therefore, we decided to use the RAID0 configuration. It is worth mentioning here, hoping to stand as a lesson for other researchers, that in the post processing work we found out that this four disk configuration was only at the limit of the required performance.

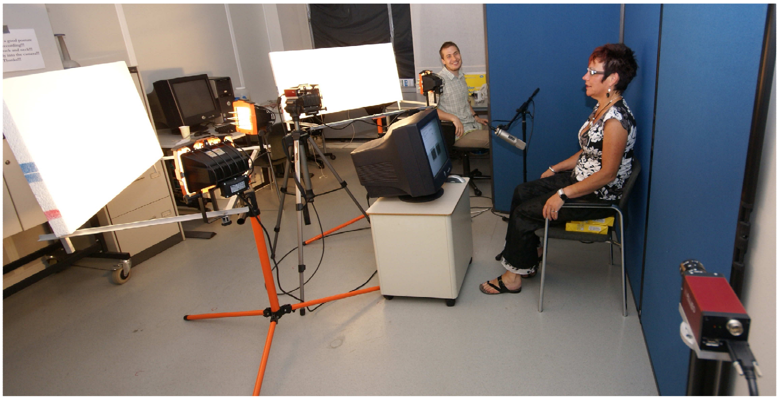


Figure 4.1: *The physical settings for the recording sessions.*

4.2.1 Video Devices

When one goes outside the range of consumer devices, things become extremely more complicated and definitely more expensive. The quality of the sensors and the huge bandwidth necessary to stream high speed video to the PC makes high speed video recording very restrictive. Fortunately, lately, by the advance made in image sensors (i.e. CCD and CMOS technology), it is possible to develop medium speed computer vision cameras at acceptable prices. We used two Pike F032C cameras built by AVT. These cameras were capable of recording at $200Hz$ in black and white, $139Hz$ when using the chroma sub-sampling ratio 4:1:1 and $105Hz$ when using the chroma sub-sampling ratio 4:2:2 while capturing at maximum resolution 640×480. By setting a lower ROI, we were able to increase the frame rate even more. Therefore, these cameras perfectly met our requirements, namely the $\frac{1}{2}$PAL resolution and $100Hz$ rate we decided upon.

4.2.2 Audio Devices

For recording the audio signal we used two high quality condenser microphones (NT2A Studio Condensers). One microphone was used for recording the background noise. We recorded a stereo signal using a sample rate of $48kHz$ and a sample size of $16bits$. The data was stored in PCM audio format. Special laboratory conditions were maintained, such that the Signal to Noise Ratio(SNR) was kept at controlled level. We considered that it is more advantageous to have very good quality recordings and degrade them in a post process as needed. The specific noise can be simulated or recorded in the required conditions and later superimposed on the clear audio data. An example of such database is NOISEX-92 ([Var93]). This dataset contains white noise, pink noise, speech babble, factory noise, car interior noise, etc. As said before, special attention was paid to the synchronization of the two modalities.

4.2.3 Side View Recordings

For recording the frontal view and side views of the speaker face, we found two solutions. The first solution we tried was to use a mirror placed at 45 ° on the side of the speaker. In this way, a parallel side view of the speaker could be captured perfectly synchronized with the frontal view. The mirror covered the speaker face entirely. In order to cope with different users, we built a wooden holder for the mirror that could be adjusted in height. The physical settings are shown in Figure 4.2. After experimenting with the mirror, we found two disadvantages of this setup. The first problem is that the side view image was distorted by the projection in the mirror, so in order to be used, this image should have been corrected in advance. The second problem we discovered was that in order to include the frontal and the side view, the field of view had to be enlarged. This would have been a step back from the point of view of our requirement to have as good detail as possible of the lower half of the face. The second solution to the dual view, and the one we actually used, is to record with two cameras, each recording its own view. The problem with this setting is that the synchronization of the two video streams is not guaranteed anymore. Fortunately, the cameras we used permitted synchronization to the FireWire bus' clock they shared. Therefore, the resulting synchronization was in the range of $125\mu s$, which is more than sufficient for our settings with $10ms$ per frame.

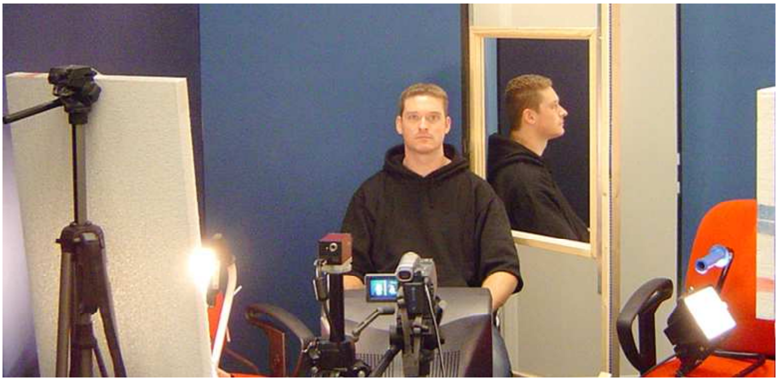


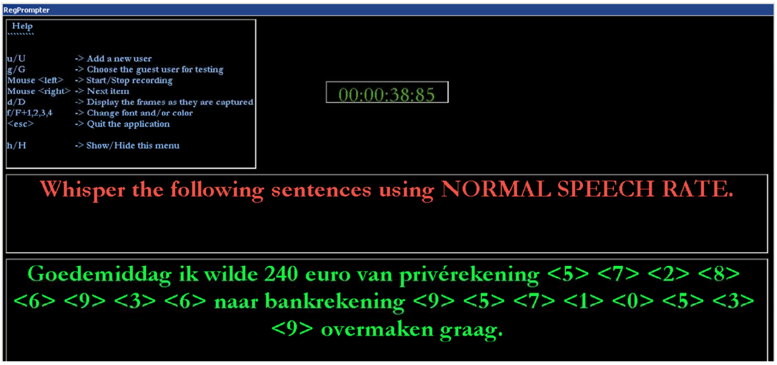
Figure 4.2: *The physical settings for the case when the side view of the speaker's face was obtained by means of a 45 ° placed mirror.*

4.2.4 Controlling the Recording Environment

The environment where the recordings are made is very important since it will determine both the illumination of the scene and the background of the speakers and the acoustics. As seen in Figures 4.1 and 4.2, we used a mono-chrome background so that by using a "chroma keying" technique the speaker can be placed in different environments making possible in this way some degree of visual noise. The windows of the room were completely blinded, the recording scene being illuminated only with artificial light. In order to remove the illumination artefacts we used two panels to diffuse the light. The acoustic background noise was recorded with a separate microphone.

4.2.5 Prompter Software

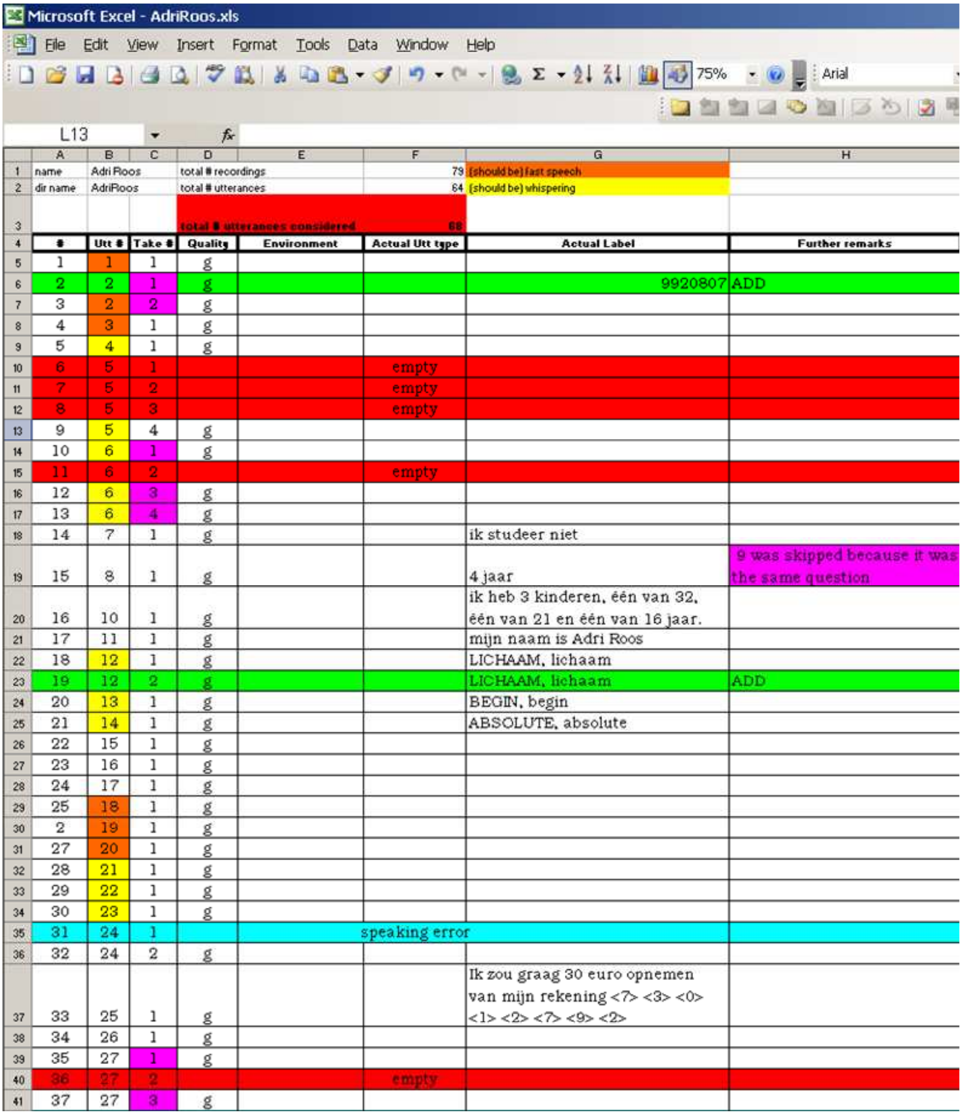
In order to control the recording devices and to instruct the speaker about the current utterance and the speaking style of the current utterance we implemented a prompter like software. The program displayed the next utterance to be uttered and the speaking style required. The user was able to start and stop the recording whenever he considered. In case when errors appeared, such as mispronunciations, word repetitions, hesitations, external noise, etc. the user had the possibility to record a subsequent take of the same utterance. The use of the prompter software greatly reduced the post-processing work. Figure 4.3 shows a print screen of the prompter. The fonts and colours were easily customisable to make the speaker's experience as enjoyable as possible. Each session lasted at least 30 minutes; therefore, there was a great stress on the speaker.



```
 Help
``````````

u/U   -> Add a new user
g/G   -> Choose the guest user for testing
Mouse <left>   -> Start/Stop recording
Mouse <right>   -> Next item
d/D   -> Display the frames as they are captured
f/F+1,2,3,4   -> Change font and/or color
<esc>   -> Quit the application

h/H   -> Show/Hide this menu
```

00:00:38:85

Whisper the following sentences using NORMAL SPEECH RATE.

Goedemiddag ik wilde 240 euro van privérekening <5> <7> <2> <8> <6> <9> <3> <6> naar bankrekening <9> <5> <7> <1> <0> <5> <3> <9> overmaken graag.

**Figure 4.3:** *The prompter tool used during recordings.*

### 4.2.6 Consistency Check During Post Processing

As we mentioned in Section 2.3.3 even in the case when the recording is automated there is still need for post processing. In our case we recorded the utterances in different folders and we automatically saved information about the speaker, the session, utterance and take. However, we still needed to check the resulting corpus for consistency. Each recording was auditioned and all the remarks were recorded. Sometimes it was needed to adjust the transcription of the recording because the speaker spoke a slightly different utterance then what was required. This was usually due to misreading, mispronunciation, some personal speaking problems or personal reflexes, etc. Other times the recording had to be completely removed from the corpus, or reversely, some utterances were recorded multiple times for various reasons. For each session each annotator produced a spreadsheet as shown in Figure 4.4.
```



	A	B	C	D	E	F	G	H
1	name	Adri Roos		total # recordings		79	(should be) fast speech	
2	dir name	AdriRoos		total # utterances		64	(should be) whispering	
3				total # utterances considered		68		
4	#	Utt #	Take #	Quality	Environment	Actual Utt type	Actual Label	Further remarks
5	1	1	1	g				
6	2	2	1	g			9920807	ADD
7	3	2	2	g				
8	4	3	1	g				
9	5	4	1	g				
10	6	5	1			empty		
11	7	5	2			empty		
12	8	5	3			empty		
13	9	5	4	g				
14	10	6	1	g				
15	11	6	2			empty		
16	12	6	3	g				
17	13	6	4	g				
18	14	7	1	g			ik studeer niet	
19	15	8	1	g			4 jaar	9 was skipped because it was the same question
20	16	10	1	g			ik heb 3 kinderen, één van 32, één van 21 en één van 16 jaar.	
21	17	11	1	g			mijn naam is Adri Roos	
22	18	12	1	g			LICHAAM, lichaam	
23	19	12	2	g			LICHAAM, lichaam	ADD
24	20	13	1	g			BEGIN, begin	
25	21	14	1	g			ABSOLUTE, absolute	
26	22	15	1	g				
27	23	16	1	g				
28	24	17	1	g				
29	25	18	1	g				
30	2	19	1	g				
31	27	20	1	g				
32	28	21	1	g				
33	29	22	1	g				
34	30	23	1	g				
35	31	24	1			speaking error		
36	32	24	2	g				
37	33	25	1	g			Ik zou graag 30 euro opnemen van mijn rekening <7> <3> <0> <1> <2> <7> <9> <2>	
38	34	26	1	g				
39	35	27	1	g				
40	36	27	2			empty		
41	37	27	3	g				

Figure 4.4: *Recordings consistency check sample.*

Because of the high speed recording and the multiple audio and video devices the bandwidth demand during recordings was extremely high. Sometimes, the system was not able to keep up with the recordings and this resulted in frames losses. During the consistency check we also checked for the number of frames lost. Some 10% of recordings was not included in the corpus we used for experiments due to a large number of missing frames. However, we still kept these recordings in the corpus, because they can still be used for other audio-visual applications.

4.3 Data Corpus Statistics

This section makes a quantitative analysis of the NDUTAVSC data corpus. The resulting corpus requires approximately 1.5TB of storage space. There were 90 recording sessions in which 70 speakers uttered a total number of 7163 utterances. The total continuous duration of the recordings was approximately 12 hours. However, due to some recording problems four speakers were removed form the final corpus. The recordings of the remaining 66 speakers are divided into 87 sessions and make up 6907 utterances which represent exactly 10 hours and 38 minutes of continuous recordings. One speaker, a female aged 52, recorded 18 sessions. She recorded 2599 utterances that account for 3 hours and 48 minutes of continuous recordings. The speaker was selected because of her clear speaking style, especially her good articulation of words. The data collected by this responded was used to build speaker dependent lip readers. Due to the time limitations we only focused on speaker dependent systems.

4.3.1 Utterance Types

We had three settings with respect to the number of utterances and their types during the recordings. We started with 64 utterances per speaker, then we recorded 125 utterances per speaker and in the end we recorded 155 utterances per speaker. The utterance types were following the recognition task introduced in Chapter 2. A complete list of all three settings is given in Appendix C. The definition of the utterance types and the number of repetitions for each type were chosen such that to optimize the amount of data and the variability of data with respect to the level of effort on the speaker side. In total 1330 digit string recordings, 1374 letter string recordings, 607 connected word recordings, 1026 banking application recordings, 333 open questions, 2570 phonetically rich random sentences were recorded. There were 211 spelling utterances 1104 whispered utterances and 207 whispered spellings. In 1080 utterances the speaker was instructed to speak using a faster then normal speech rate. In the case of our long run respondent we collected 712 digit string recordings, 862 letter string recordings, 9 random word sequences, 4 open questions and 901 phonetically rich random sentences. She recorded only 3 spellings, 16 whispered utterances and 3 whispered spellings. In 16 cases she was requested to speak at faster then normal speech rate.

4.3.2 Respondents

It is very difficult to guaranty the complete coverage of the speakers' variability. The directions that need to be taken into account are: gender, race, speaker dialect, age distribution, education level.

We recorded 70 respondents. However, as mentioned above, the data recorded for four of them had to be removed; three due to some hardware problems during recordings and one due to the archive damage during storing. From the remaining 66 respondents 62 were native Dutch speakers and four had various nationalities but were speaking with high proficiency. Two other speakers were bilingual. Due to the location of the experiments, the first option was to ask our colleagues to take part in the recording sessions. This resulted into a very biased distribution with respect to age, gender and education level. In order to correct this bias and balance the pool of respondents we asked the administration staff which is in majority females. We obtained in this way a ratio of 46 to 21, male speaker being still more numerous. The subjects were aged 19 to 64 but as seen in Figure 4.5 the age is still biased towards the range 19 to 28 years. Figure 4.5 shows the distribution of speakers with respect to age and gender. Even though we did not cover completely the speakers' variance, we managed to record a relatively large number of speakers and we consider that we made an important step towards building of a strong data corpus. We envision that based on the recording settings we developed, in the future the current database will be upgraded such that to completely correct the current issues.

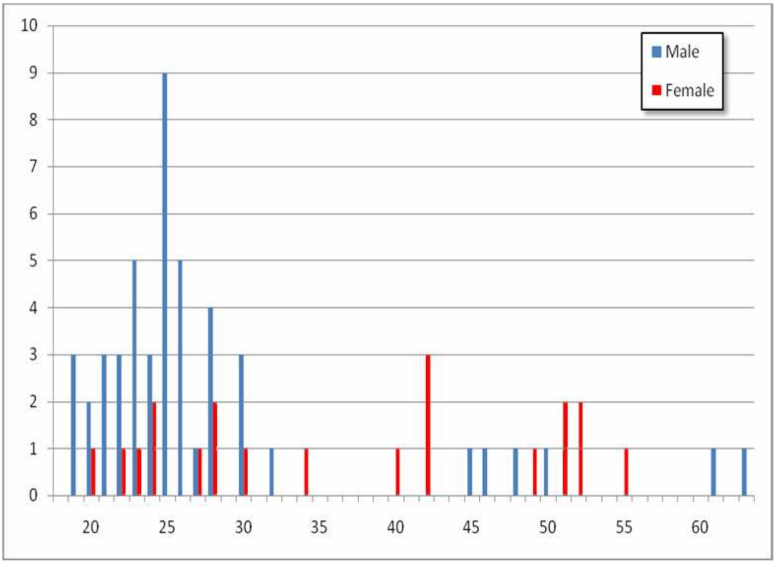


Figure 4.5: *Age and gender distribution of the speakers from the NDUTAVSC corpus.*

4.3.3 Language Data

As we mentioned in Chapter 2 the language data was build starting from the language data used in the DUTAVSC corpus and this corpus contained among others a selection of utterances from the Polyphone corpus [Boo94]. The language data was divided in 7 sets as follows: the digits 0 to 9, the 26 letters of the Dutch alphabet, 1966 random words, 1108 phonetically rich sentences, 72 conversation starters and endings, 91 context aware sentences (i.e. banking applications sentences) and 41 simple open questions (i.e. for these questions the user was asked to answer the first thing that came to mind. In this way we expected to collect spontaneous speech utterances). The utterances presented to the speakers were randomly selected from the utterances sets in accordance with recording protocol shown in Appendix C. The final data corpus contained 60399 recorded words made up from 2605 unique words. The long run respondent recorded 17997 words made up from 1864 unique words. Since we had a relatively large number of words recorded the distribution of the number of samples from each word is highly skewed. The most recorded words were the digits(e.g. the most recorded digit was “9” which was recorded 1805 times) followed by the short words like “de” (the second most recorded word with 1748 times) , “ik”, “van”, “een”, “mijn”, “op”, “het”, “in”, and the letters of the alphabet. Also a very high occurrence have the words related with the banking applications like “bankrekening”, “storten”, “rekening”, “overmaken”, “wilde”, “wil”, and others. A similar situation was found for the recordings made for the speaker dependent purposes. In this case the most recorded word was the word “de” that was recorded 828 times. The word “de” is the most used from the two definite articles in Dutch, “het” is the other one, which justifies the fact that it is the most recorded word. The direct consequence of recording real sentences is that the distribution over the number of samples per words approximates the real distribution over the usage of the words in the language. Figure 4.6 shows the number of recordings of the first 200 most recorded words. The words with a rank larger then 200 appear less than 20 times in the corpus. Figure 4.7 shows the distribution for the speaker dependent case. However, as the digits and letter strings are one of the most used recognition tasks in lip reading resulted in their appearance being more prominent in our data and, therefore, in a artificial distribution of the words. Figures 4.8 and 4.10 show the counts for the digits, while Figures 4.9 and 4.11 show the counts for the letters. The representation of the classes in the training data influences the performance of the classifier. If any class is significantly more present in the data corpus, then the model statistics can be biased towards that specific class and could yield faulty results. On the other hand, we think it is important that each word be present in the data corpus corresponding to its probability of appearance in the recognition task. One more aspect to observe here is that the different recognition tasks are differently represented in the corpus. This can also explain some of the differences in performance of the corresponding recognisers.

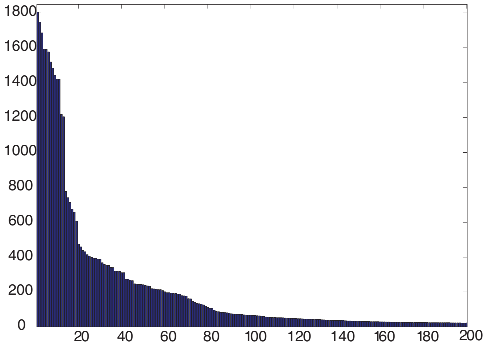


Figure 4.6: *The number of takes of the first 200 words as appear in the NDUTAVSC corpus.*

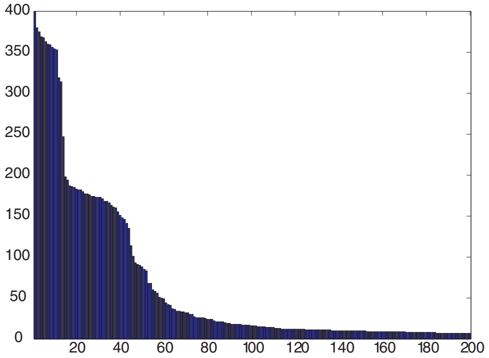


Figure 4.7: *The number of takes of the first 200 words as appear in the speaker dependent NDUTAVSC sub-corpus.*

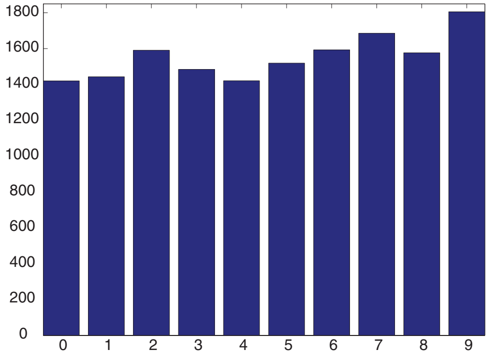


Figure 4.8: *The digit distribution in the NDUTAVSC corpus.*

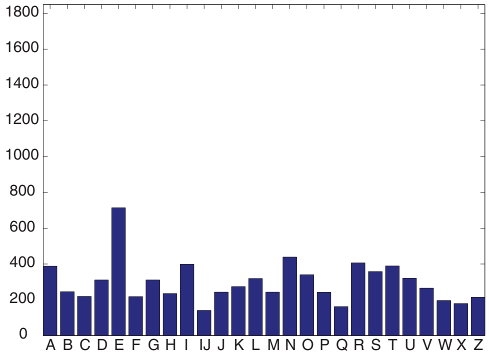


Figure 4.9: *The alphabet distribution in the NDUTAVSC corpus.*

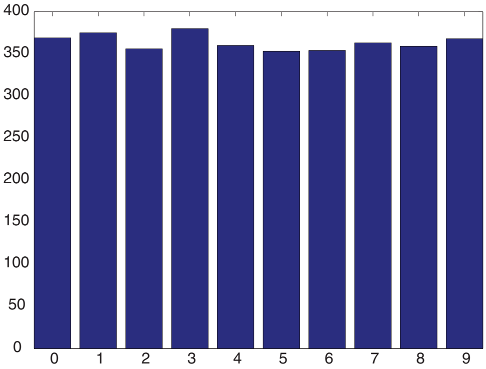


Figure 4.10: *The digit distribution in the speaker depended NDUTAVSC sub-corpus.*

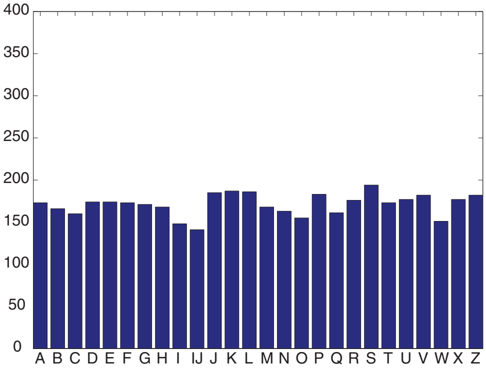


Figure 4.11: *The alphabet distribution in the speaker depended NDUTAVSC sub-corpus.*

4.3.4 Speech Rate

As shown in Appendix C we asked the respondents to change their speaking style during the recordings. According to the recordings protocol, a number of 1080 utterances should have been spoken using a high speech rate. However, we found out that while some of the speakers were not able to speak fast others spoke very fast even in normal situations. This is not something out of the common sense expectations, but forced us to carefully reclassify the utterances.

The speech rate computed for the entire data corpus has minimum of 14WPM and maximum of 318WPM with the mean around 93WPM and a standard deviation of 45WPM. For the recordings that were marked as slow in the recording protocol we computed a maximum of 252WPM speech rate and a mean of 84 with the standard deviation of 39WPM. On the other side, the minimum speech rate for the recordings marked as fast in the protocol that we found was 49WPM, which is relatively small. The mean was, however, 142WPM which is considerably larger then the 84WPM value found in the low speech rate case. The standard deviation is relatively large (i.e. 44WPM) since some speakers were not able to sustain a fast speech rate.

We decided that 160WPM is a better threshold for deciding on the classification of the utterances with respect to the speech rate. In this case, we found that the mean speech rate is around 188WPM with a standard deviation of 27WPM. We also considered the definition of the word based on 5-letter multiples which takes into account the length of the words. Table 4.1 gives the computed values for 5-letter multiples words definition and for the regular word definition for the entire corpus, but also for the sub-corpus with the speaker we used for speaker dependent analysis.

In the case when the classification of the utterances in high speech rate versus normal and low speech rate, we found that instead of 1080 utterances we can actually consider as high speech rate with only 519 utterances. However, when the 5-letter word definition was used, this number became 2372. The same situation was found for the speaker dependent recordings as well, namely instead of 16 utterances with high speech rate we found 25 and 504, respectively. Table 4.2 shows the actual coverage of the declared high speech utterances and the found high speech rate utterances.

Table 4.1: *Speech rates in the NDUTAVSC corpus.*

	Minimum	Maximum	Mean	Std
Entire corpus: normal word definition				
global	14	318	94	46
slow	14	252	85	40
fast	49	318	143	45
> 160WPM	161	318	188	28
Entire corpus: 5-letter word definition				
global	14	398	128	71
slow	14	377	114	62
fast	58	398	206	62
> 160WPM	161	398	208	40
Sub-corpus: normal word definition				
global	14	241	70	38
slow	14	216	70	37
fast	88	241	164	43
> 160WPM	161	241	182	21
Sub-corpus: 5-letter word definition				
global	14	343	90	61
slow	14	277	89	60
fast	131	343	242	66
> 160WPM	161	343	183	23

4.3.5 Viseme Coverage

The high speed recordings were justified by the need to ensure the coverage of each viseme by a sufficiently large number of frames. It was shown in Section 2.3.2 that this is especially important in the case when the speech rate is high. In the

Table 4.2: *The declared high speech recordings versus the actual high speech recordings in the NDUTAVSC corpus.*

Declared	Actual	Covered	Percent covered
Entire corpus: normal word definition			
1080	519	354	32.78%
Entire corpus: 5-letter word definition			
1080	2372	798	73.89%
Sub-corpus: normal word definition			
16	25	10	62.50%
Sub-corpus: 5-letter word definition			
16	504	13	81.25%

case of the old data corpus the speakers were only using their normal speech rate and no requirements were made in order to increase the speech rate. Even in this case the range of the speech rate was very high. As we have seen in the previous section in the new data corpus, the speech rate range has increased even more due to the fact that we asked the speakers to change their speech rate during recordings. It is interesting to note that the range has enlarged in both directions towards 0 and towards infinity. This is probably due to the fact that sometimes the speakers tended to speak slower than their normal speech because they intended to make a clear distinction between their fast speech rate and their normal speech rate. When we selected the fast speech rate utterances, the increase is very visible in all cases, the question is whether the 100Hz frame rate was necessary or not. Table 4.3 shows the statistics for all considered cases. It is clearly visible that in the case of normal speech rate the mean number of frames per visemes is relatively high and is definitely higher than in the case of the old data corpus. However, while the frame rate has increased 4 times the mean FPV has only increased 2 times in the global analysis and 3 to 4 times in the slow speech case. In the case of high speech rate we found that the mean FPV values are concentrated around the value 8 in all cases. The variance of the coverage is extremely low compared with the normal speech case. That is the case because the normal speech rate goes from very slow to the upper limit, while the fast speech rate is conditioned to the extreme values of a normal speaker. The increase in coverage is again consistent with the increase in frame rate, namely from a mean of 3FPV in the DUTAVSC corpus to 8FPV in the case of the new corpus, however, as in the general case the increase is slightly less. Figure 4.12 shows the histogram of the coverage for the entire corpus when all utterances are considered. Figures 4.13 and 4.14 show the histograms for the high speech rate for the case of the declared high speech utterances and the case of the real high speech utterances (i.e. the ones that have more then 160WPM). The shift to the right of the distribution is clearly visible. The situation is similar in the case of the sub-corpus. It is worth noting that we have found 8 utterances which had a FPV value equal to 4 and that approximately 900 utterances had a FPV value less than 8. These utterances would have had less than 1FPV and less than 2FPV, respectively, if the 25Hz standard frame rate would have been used.

The conclusion here is that in many cases the high speed recordings are a necessity and the distinction between high speech rate and low speech rate needs to be addressed when building a robust lip reader.

Table 4.3: *The viseme coverage by data, computed as the number of Frames Per Viseme (FPV) in the NDUTAVSC corpus.*

Minimum	Maximum	Mean	Std
Entire corpus: global statistics			
4	167	22	23
Entire corpus: slow speech			
4	167	25	24
Entire corpus: declared fast speech			
4	26	8	3
Entire corpus: slower than 160WPM			
8	167	30	25
Entire corpus: faster than 160WPM			
4	14	8	1
Sub-corpus: global statistics			
4	167	36	31
Sub-corpus: slow speech			
6	167	36	31
Sub-corpus: declared fast speech			
4	10	7	2
Sub-corpus: slower than 160WPM			
9	167	42	31
Sub-corpus: faster than 160WPM			
4	11	9	1

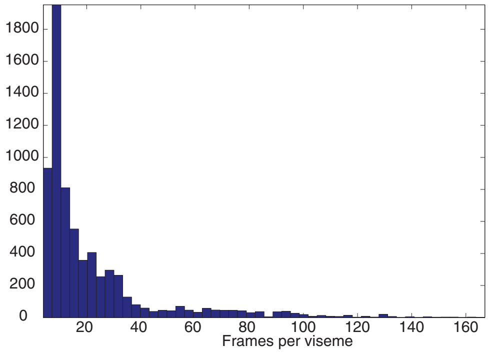


Figure 4.12: *The viseme coverage in the NDUTAVSC corpus. The frame rate is 100HZ and all the utterances are included.*

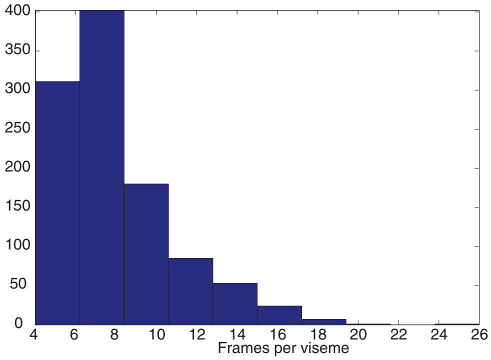


Figure 4.13: *The viseme coverage in the NDUTAVSC corpus when only the utterances declared as fast speech are analysed. The frame rate is 100Hz.*

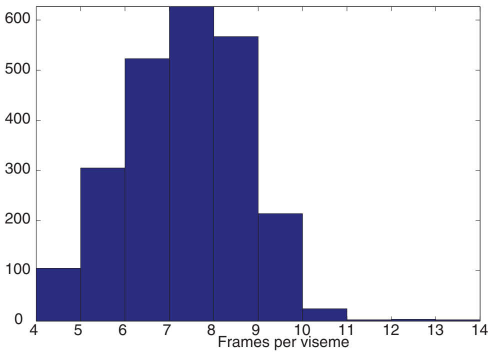


Figure 4.14: *The viseme coverage in the NDUTAVSC corpus when only the utterances that are spoken at more than 160WPM speech rate are analysed. The frame rate is 100Hz.*

4.4 Discussion

In this chapter we presented the data corpus build for this thesis. Building this corpus took an important slice from the time dedicated for the research on which this dissertation is based on. However, as we argued in Chapter 2 and as it emerged from the analysis made in this chapter, this work was necessary. In general, the data acquisition process is very important for the success of the research endeavour, but many times it is done improperly due to the fact that it is extremely time consuming and it is not always possible to valorise it from the scientific point of view (i.e. it is not as strong from the point of view of its scientific content and therefore it is not very easy to publish scientific papers).

We presented in this chapter the recording settings used during the experiments and analysed the data that resulted afterwards. We included exact statistics on the resulting corpus and compared the new corpus with the old one. Even though we did not completely cover all the requirements, we decided for the corpus, such that to have a balanced corpus with respect to the speaker characteristics, and to have a language coverage comparable to the case of the speech recognition corpora. Nevertheless, we build a very large corpus, to our knowledge this is the largest corpora for lip reading to date of this writing, and we are confident that this corpus can be a very good starting point. In any case, as it was the case for our research, this corpus can be used for testing different approaches.

Appendix D lists an example of the utterances recorded as they were presented to the speaker through the prompter software.

Chapter 5

Statistical Lip Geometry Estimation for Lip Reading

In this chapter we present a method for describing the shape of the mouth based solely on a statistical interpretation of the distribution of the pixels that lie on the lips and the performance obtained by lip reading systems which are trained based on the features vectors defined using this method. This method is special because it does not rely on an a-priori defined model of the lips but still is not a bottom-up approach since it does not try to detect any special features of the mouth (i.e. it does not search key points or other low level features). The method was first described by Wojdel and Rothkrantz in [Woj00].

The chapter starts by introducing the details of the algorithm giving as well some examples of image filters that could be used in the preprocessing of the input image. Thereafter, it gives some methods for making the algorithm more robust for instance by restricting the ROI and by refining the output of the filter through an outlier detection method. After giving some validating examples for the suitability of the method, the end of the chapter presents the performance of the lip reader built based on this method.

This chapter presents as well a method for the detection and the description of the teeth, tongue and cavity. These elements are very important ([Wil98a]) for lip reading and can be used as extra features in any settings, irrespective to the main feature extraction technique. They are introduced in this chapter because the detection algorithm is similar with the main topic of the chapter.

5.1 Description of the Algorithm

This method uses an image filter to compute the probability of each pixel to be part of the lips. The result of the filter is then statistically interpreted in order to describe the shape of the mouth. The analysis of the shape of the mouth is independent of the choice of the image filter, however, the performance of the algorithm is strongly linked with the capacity of the filter to exactly identify the pixels that lie on the lips. An overview of the algorithm is depicted in Figure 5.1. The algorithm evolves as a processing pipeline and contains the following steps: extraction of the next frame, ROI detection or tracking, application of an image filter to describe the pixels, analysis of the filter result and refinement, and finally analysis of the refined filter result and computation of the output features. The following sections present in detail all the necessary steps to obtain the depicted images.

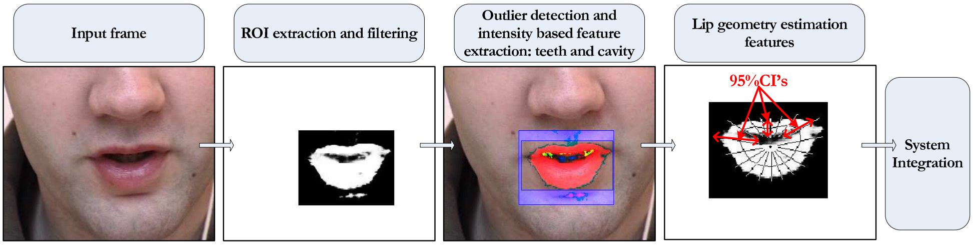

Input frame
ROI extraction and filtering
Outlier detection and intensity based feature extraction: teeth and cavity
Lip geometry estimation features
95%CI's
System Integration

Figure 5.1: *The process of estimating the lip geometry shown on real example. From left to right we have the input image, the result of the ROI detection and image filter procedure, the outlier removal and the lips, cavity and teeth detection and finally the geometric features shown in polar co-ordinates.*

5.1.1 Defining the Region Of Interest

As the first step of the processing pipeline we have to locate the face and then the mouth of the speaker. The reduction of the searching area (i.e. ROI) removes unnecessary parts from the image which is very important from at least two reasons: first the processing time is greatly reduced and secondly many possible unwanted artefacts can be avoided. For this we used the Viola-Jones algorithm for object detection [Vio01]. This classifier uses a new method for detecting the most representative Haar like features using a learning algorithm based on AdaBoost. It combines a set of weak classifiers using a "cascading" approach which corroborated with a fast method for computing the Haar-like features, allows for high speed and very low false-negative rates. In order to increase the reliability of the ROI extraction process we used a combination of detection and tracking steps. Hence, in a first step the mouth of the speaker is detected using the Viola-Jones detector, and in the subsequent steps the mouth is tracked using an object tracking algorithm which is adapted using the last detected ROI. The object tracking algorithm uses a Gaussian Mixture Model to model the colour distribution of the object and of the background and a deformable template to optimally fit the tracked object.

5.1.2 Lips Segmentation

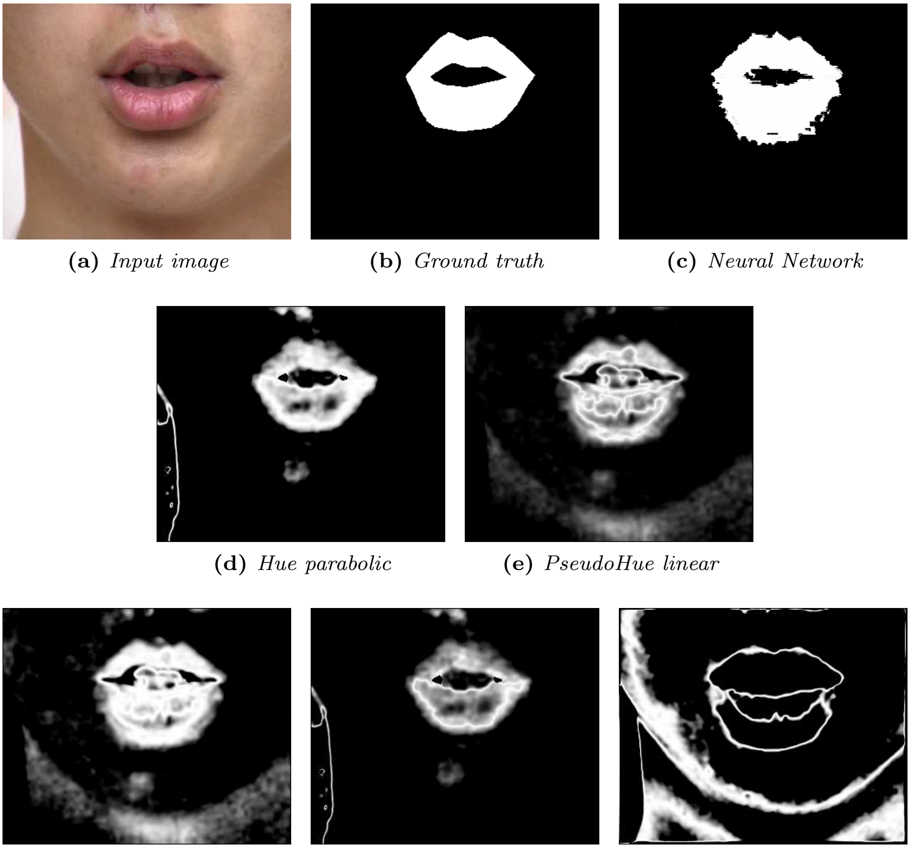
The next step in the process is to somehow compute the probability of each pixel to belong to the lips. Fortunately, because the input image contains only the mouth area and since the lips have a distinct colouring we can extract the lip's pixels without the need for complicated additional object recognition techniques. The *lip-selective* filter is not fixed to any pre-chosen image filter and therefore any available method can be used. The only requirement is that the filter returns the result in probabilistic terms, namely, each pixel should be given a value between 0 and 1, 1 meaning maximum confidence that the given pixel belongs to the lips. As discussed in Section 2.4.4 it is very important to choose the most relevant colour encoding system. We tested several image filters based on different colour systems. As with all image segmentation techniques the illumination conditions and the quality of the recorded video sequences greatly influences the end result. A parabolic shaped filter is very simple and from the point of view of computing power requirements very attractive. Unfortunately, during our experiments we found that in many cases, if the illumination of the face is not perfect, the hue component itself is not sufficient for proper lip selection. Even combining in a cascade the results from parabolic thresholding in different colour spaces did not yield sufficiently robust filters. On the other hand, we found that training a simple feed-forward neural network was performing much better for our data corpus. The network that was used has only 5 neurons in a hidden layer and one output neuron. The network was trained using directly the RGB values from the pixels in the image. This filter achieved extremely accurate results. The results obtained with several filters are shown in Figure 5.2, where it is visible that the neural network outperforms all the other approaches. In all situations the same area was used for training, namely the area covering the lower lip. One problem of the hue based filter is caused by the fact that the hue is not well defined for very bright areas. This can be seen in the figure in the lower left of the corresponding images where a very bright area is present. The hue filter generates here an erroneous edge. The pseudo-hue filter has on the other hand, problems with the dark areas, therefore, the shadows, or in our case the mouth cavity, tend to appear in the filter's results. The filter based on the luminance is failing to 'see' the lips, however, the bright edges seem to be detected very well. Comparing the thresholding approach with the parabolic approach we can conclude that it does a better job in assigning the probability.

5.1.3 Defining the Feature Vectors

Here is where the innovation of this approach comes into action. The result of the filter is considered to be the distribution function $I(X, Y)$ of a spatial bivariate distribution. The first observation to be made is that the expected location of this distribution: $[EX, EY]$ accurately approximates the centre of the mouth. Using the mean location we transform the filter's result into polar coordinates using the following formula:

$$J(a, r) = I(EX + r\cos(a), EY + r\sin(a)). \tag{5.1}$$

The algorithm estimates the shape of the mouth by describing the conditional distribution, conditioned on the direction, obtained from the distribution function J. For each angle α we, therefore, define the mean and the variance of the conditioned distribution using the following formulas:



(a) *Input image* **(b)** *Ground truth* **(c)** *Neural Network*

(d) *Hue parabolic* **(e)** *PseudoHue linear*

(f) *PseudoHue parabolic* **(g)** *Hue linear* **(h)** *Luminance parabolic*

Figure 5.2: *The results of several image filters used to segment the lips pixels.*

$$M(\alpha) = \frac{\int_r J(a,r)rdr}{\int_r J(a,r)dr}, \; \textit{and} \tag{5.2}$$

$$\sigma^2(\alpha) = \frac{\int_r J(a,r)(r - M(\alpha))^2 dr}{\int_r J(a,r)dr}. \tag{5.3}$$

As an image is discrete rather that continuous, all of the values are obtained from summation rather than integration, so we only operate on estimations of those values, namely $\widehat{M(\alpha)}$ and $\widehat{\sigma(\alpha)^2}$. For each given angle the conditioning is defined as the circle sector which contains in its centre the vector from the mouth centre which makes the angle α with the horizontal. The rightmost image from Figure 5.1 shows an example of the estimations of the two values for a number of 18 directions. In this image the values of $\widehat{M(\alpha)}$ are shown for each angle and are linked together by a line. It is clearly visible that this polygon passes through the centre of the lips and accurately describes the shape of the mouth. On the other hand, for each angle the perpendicular lines depicted in Figure 5.1 represents the 95% confidence interval for the conditional distribution, namely it describes the range $[\widehat{M(\alpha)} \pm 1,96 * \widehat{\sigma(\alpha)^2}]$. So it is obvious that the two sets of values accurately describe the shape of the mouth. The 95% confidence interval clearly describes how thick the lip in that specific direction is, while the means describe the shape of the mouth. We should note that the lips of a wide-stretched mouth, appear thinner than those of a closed mouth when related to the overall size of the mouth.

The accuracy of the procedure depends on the performance of the image filter used. The difficulties on the filter's side come from the artefacts that can appear on the speaker's face: shadows, areas on the face that have the colour as the lips, the use of coloured lipstick, etc.; some of these problems have been diminished by using an outlier removal technique on the filter's result. This is presented in Section 5.1.5.

The feature vectors are defined as the vectors containing the combined sets of values, computed for a number of angles, after the appropriate normalization. Choosing the number of directions is a compromise between accuracy and processing efficiency. The longer the vectors, the more information on the original distribution they contain but the longer it takes to extract and process them. Also higher dimensionality generally makes it more difficult to train the recognition modules. Wojdel and Rothkrantz indicate that a division of the space into 18 sectors is optimal for obtaining good results ([Woj00]). Therefore, we used 18 sectors, shown in Figure 5.3, to estimate the shape of the mouth, resulting in feature vector of dimension 36.

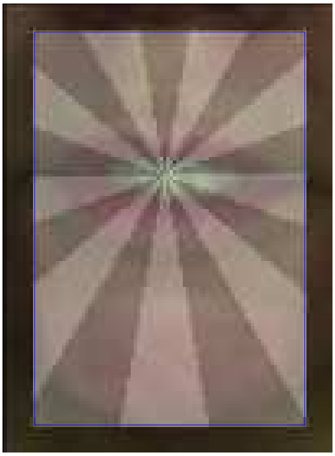


Figure 5.3: *The 18 feature sectors centred in the centre of the mouth.*

5.1.4 Visual Validation of the Feature Vectors

Figure 5.4 shows a sequence of frames with the final annotations and the corresponding features extracted. The feature vectors extracted from the entire recording using the above approach are depicted in Figure 5.5. The beginning and the end of the utterance are clearly visible as indicated on top of the image. The longer pauses between the words are also slightly visible. Also visible are the round areas located in the variance part which represent the time when the mouth was widely opened. Starting from this image alone we can conclude that there is significant speech related information in the resulting feature vectors.

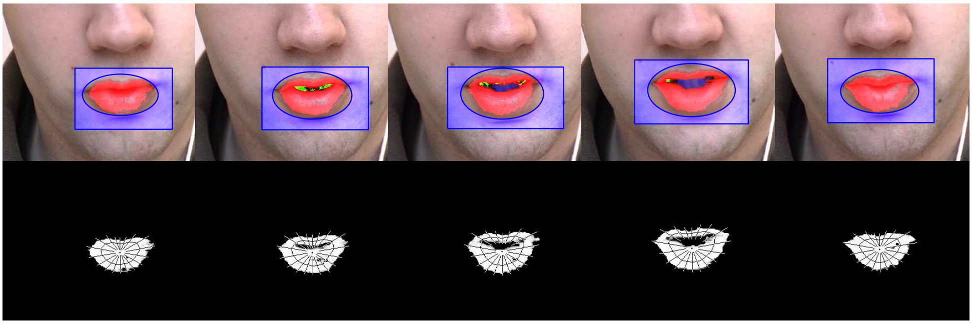


Figure 5.4: *Example of processed frames. The annotation and the extracted features are shown as well.*

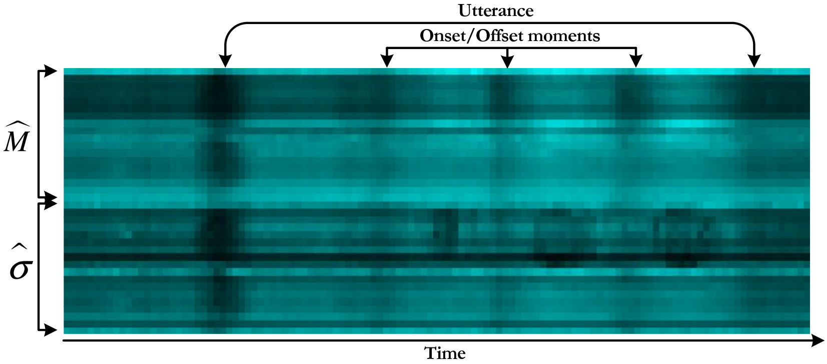


Figure 5.5: *Pairs of* $\widehat{M(\alpha)}$ *and* $\widehat{\sigma(\alpha)^2}$ *vectors extracted from a video sequence. The beginning and ending of the utterance as well as the pauses between the consecutive words are marked by arrows.*

Figure 5.6 shows the sample mean of $\widehat{M(\alpha)}$ computed over all utterances that contain the letters A, H and K, respectively. The viseme transcriptions for the three letters are as follows: A = [aa], H = [h aa] and K = [gkx aa]. We can see that in this figure the shapes corresponding to the letters A and K are most of the time overlapping making the classification somewhat difficult. However, as seen in Figure 5.7, they become more distinguishable when considering the values for $\widehat{\sigma(\alpha)^2}$. On the other hand, the situation is almost reversed when analysing in the two images the shapes corresponding to the letters H and K. Therefore, this is an example of a situation when the values of one feature type are not sufficient for discriminating among the three letters. We should remark here that, because in the image 5.6 the graph contains the mean of each feature for each angle, we see in fact the mean shape of the mouth during uttering of the corresponding letter. Therefore, this image is not completely reliable since the information about the dynamics of the mouth during speaking are lost due to averaging. Figure 5.8 shows the standard deviation of $\widehat{M(\alpha)}$ which brings back some information about the mouth dynamics.

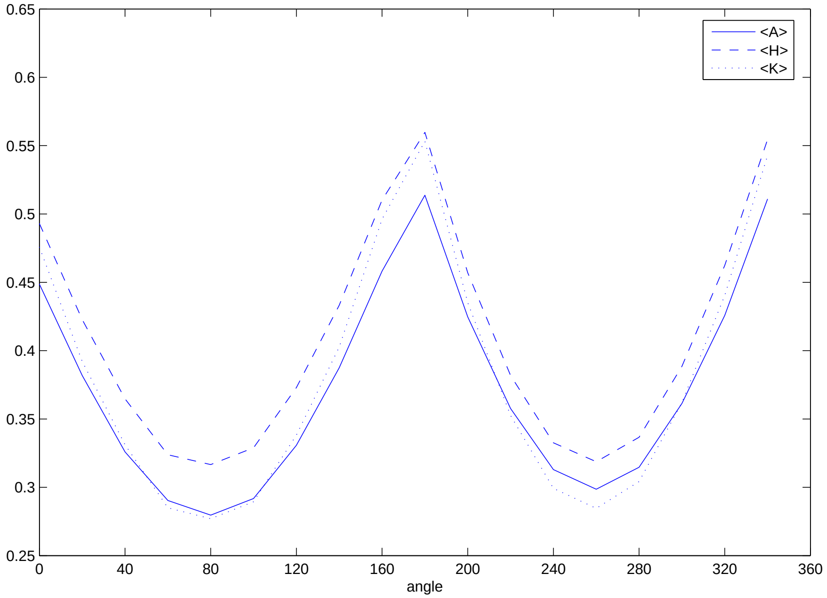


Figure 5.6: *The average of* $\widehat{M(\alpha)}$ *computed over all utterances containing the letters A, H and K, respectively.*

5.1.5 Refinement of the Filter Results: Outlier Removal

As can be seen in the second and the third images in Figure 5.1, even after reducing the area of interest and even with optimal filtering of the mouth in some cases, the filtered image still contains unwanted artefacts. In these images large artefacts are visible for instance just below the lower lip. This can be the case when there are areas on the speaker's face of which colour exactly matches the colour of the lips, such as wounds, acne vulgaris or other marks on the face. In order to reduce the impact of such occurrences a process of outlier detection and deletion can be used before the actual feature extraction. The blue area shown in the third image of Figure 5.1 but also in the first row of images in Figure 5.4, superimposed on the input image, represents the area which is affected by the outlier deletion process. The algorithm uses the same interpretation of the filter's result, namely as a spatial distribution function, and defines an outlier as an outlier of this distribution. Therefore, an outlier is defined as any location that is further away from the mean by more

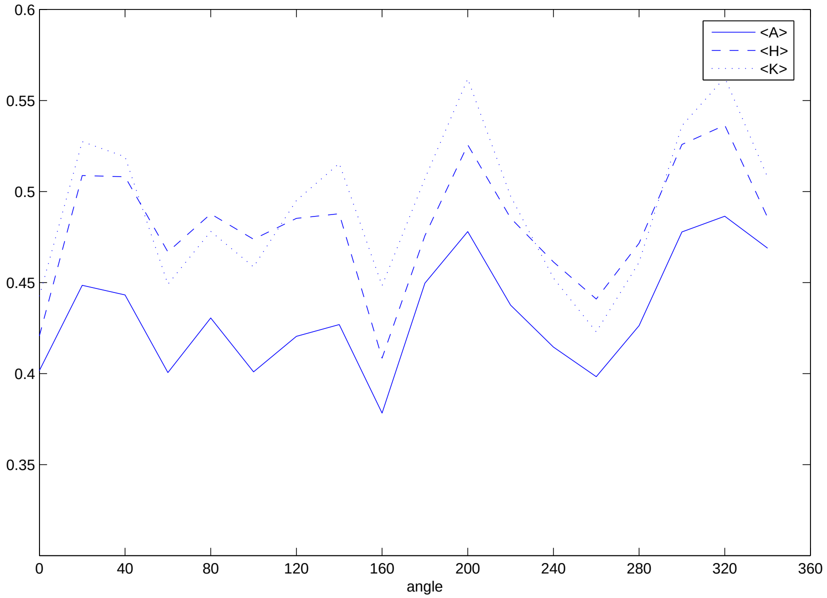


Figure 5.7: *The average of* $\widehat{\sigma(\alpha)^2}$ *computed over all utterances containing the letters A, H and K, respectively.*

than a number of standard deviations. The outlier detection process assumes that the image filter, even though it is not perfectly accurate, still assigns the correct probability to a significant number of lip pixels. In Figure 5.1 we used a rectangular delimitation area, which makes use of a-priory knowledge about the shape of the mouth. Therefore, we applied a different definition of an outlier on the horizontal axis than on the vertical axis. More accurate outlier detection is obtained by defining elliptical delimitation area as shown in Figure 5.4. The rectangular/eliptical shape refers to the interior edge of the blue area in these images. In Figure 5.9 we show an example where the necessity of an outlier detection scheme becomes clearer. In this figure in the images in the left we have the results when no outlier detection is used, while on the right we have the results when elliptical outlier detection is used. We can see that the impact on the resulting features can be very large.

5.1.6 Cavity, Tongue and Teeth Description

The shape of the lips is not the only determinant of a spoken utterance. It was shown that the position of the tongue and the appearance of the teeth while lip reading are important sources of information for the human lip reader. However, these elements are not always visible and sometimes their visibility is not only correlated with the spoken utterance but also with the speaking style of the interlocutor. It is essential in the case of lip reading to extract from the visual channel as much information as possible about the utterance being spoken, especially given the fact that compared

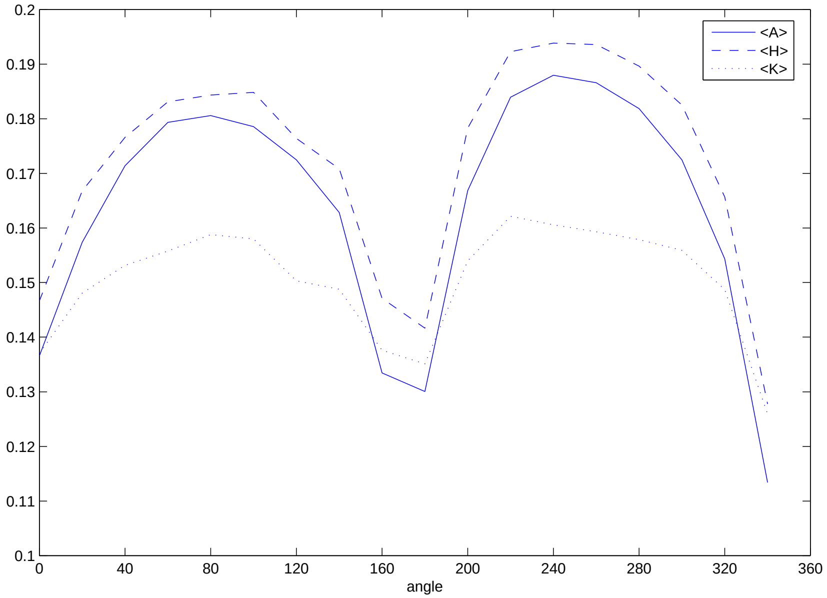


Figure 5.8: *The standard deviation of $\widehat{M(\alpha)}$ computed over all utterances containing the letters A, H and K, respectively.*

with the audio modality it is agreed that the visual speech inherently provides less information. We proposed that this type of information should be used to enrich the feature vectors in any case. Tracking of the teeth is easier than tracking the tongue. The teeth are much brighter than the rest of the face and can therefore be located using a simple filtering of the image intensity. The visibility and the position of the tongue cannot be determined as easily as in the case of the teeth, because the colour of the tongue is almost indistinguishable from the colour of the lips. We can, however, easily determine the amount of mouth cavity that is not obscured by the tongue. While teeth are distinctly bright, the whole area of the mouth behind the tongue is usually darker than the rest of the face. So we can apply an intensity based thresholding filter for both cases. The teeth and cavity areas are both highlighted in Figures 5.1 and 5.4, the cavity in green and the teeth in dark blue. In order to describe the appearance of these two elements into quantitative data we used the total area of the highlighted region and the position of its centre of gravity relative to the centre of the mouth. Therefore, the feature vectors were enlarged with 6 more entries for every frame.

5.2 Lip Reading Results

We used the algorithms described in this chapter to process our data corpus. For each recording a file was produced containing one feature vector for each video frame in the recording. Each vector had 42 components, 2 × 18 shape features, namely

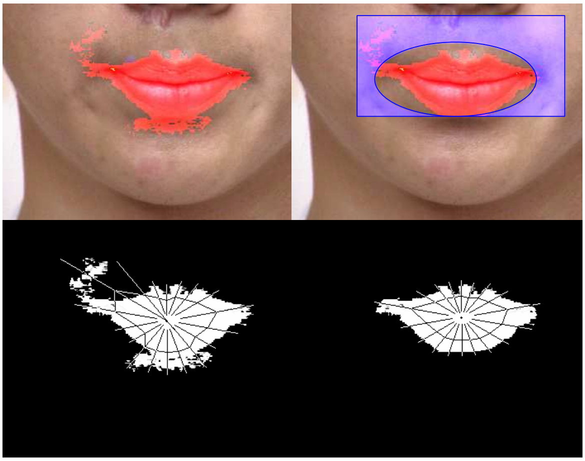


Figure 5.9: *Outlier detection for improving the estimation of the lip geometry.*

for 18 angles the mean and variance of the conditional distributions and 6 intensity features describing the presence of the teeth and the cavity. We trained a lip reader based on the HMM approach for each recognition task described in Section 2.2, namely for connected digits, connected letters, grammar based utterances, random sentences and the complete corpus. The best result was achieved for the digit recognition task for which we obtained 43.24% word recognition rate and 33.59% accuracy rate. As discussed in the Chapter 2, there are several ways to improve the basic performance of the system. We, therefore, considered the inclusion of the first and second derivatives of the feature vectors. This meant that the new feature vectors would have 126 entries. In the first group of settings each state of the HMMs used for inference contains a 42-dimensional Gaussian, while in the second group of settings this changes to a 126-dimensional Gaussian. Therefore, the number of parameters that need to be computed during the training stage is increasing many folds with the new settings. If there is enough data in the training set to train all the new parameters, the performance of the new recognisers increase considerably. In our case we found that for the best case, namely the digit recognition, the word recognition rate was 54.05% and the word accuracy was 43.24%. In the case of the letter string recognition task the increase was smaller, from 29.48% word recognition rate to 34.33%. We think that this can be explained by the fact that the size of the corpus used for the letter recognition task was smaller than in the case of the digit recognition task. In each state the distribution over the observed features is approximated using a Gaussian distribution. However, the real distribution of the observations is rarely Gaussian. Therefore, in order to make the approximation better, a mixture of Gaussian distributions is used. Since the number of Gaussian distributions suitable to approximate the real distribution is not known in advance, a trial and error approach is usually used. In our case this approach proved to be very useful since the performance of the recognizer was substantially increased. For instance in the case of the digit recognition task, when only the static features were used we obtained a word recognition rate of 58.69% and word accuracy of 49.81%. This result was obtained when 32 Gaussian mixtures were used. In the same experiments, but when the dynamic features were added, the results increased as well to a maximum of 74.52% word recognition rate and 59.46% word accuracy with 25 Gaussian mixtures used. All the results above were obtained from systems that considered isolated visemes as building blocks. This means that no influence is considered from neighbouring visemes. However, in speech recognition there is a good practice to use the left and the right viseme to build a local context as it was introduced in Section 2.6.3. This is only possible when there is sufficient data in the training corpus so that the number of unseen context combinations is very low. In the case of continuous speech recognition tasks this is very difficult to obtain. However, we were able to use this approach for the digit and letter recognition tasks with great success. For instance, when the full feature vectors combine the static and the dynamic features, the increase in word recognition rate is from 54.05% to 69.88%. When combining all the above approaches we obtained for the digit recognition task the best results when using a 28 Gaussian mixture, namely a word recognition rate of 89.96% and a word accuracy of 76.83%. Figure 5.10 shows the graph of word recognition rate and words accuracy as a function of the number of Gaussian mixtures for the case of the CD task where all the above tuning was used. It is clearly visible in this case that having sufficient data for training using a Gaussian mixture to approximate the state distribution has a great impact.

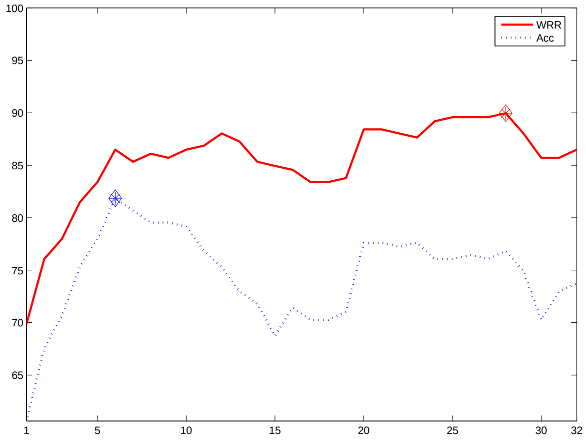
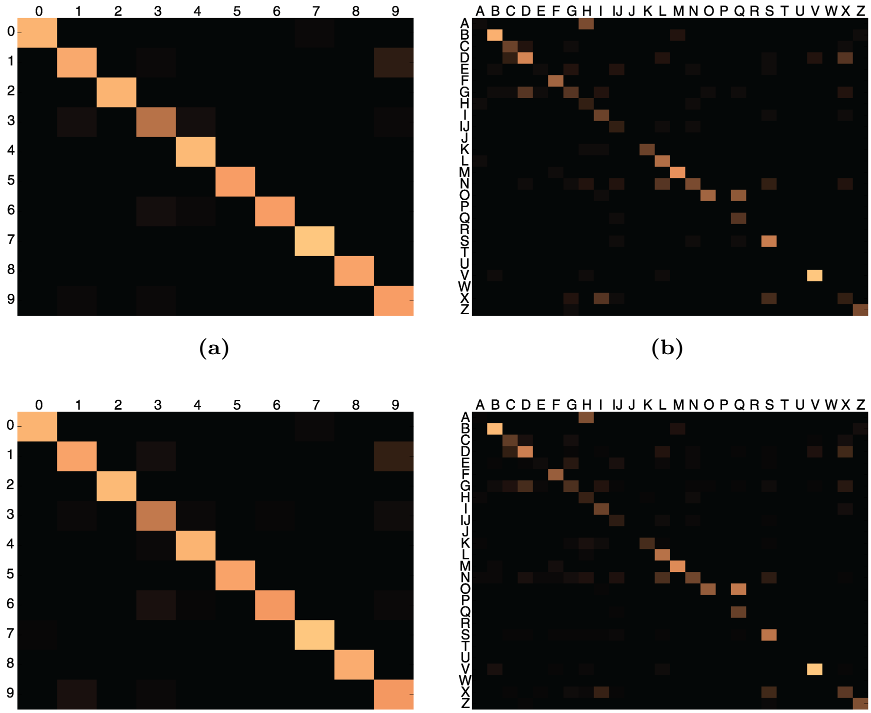


Figure 5.10: *The WRR and Acc results for CD recognition task as a function of the number of mixtures. The x axis gives the number of mixtures and the y axis shows the results obtained. The recognition system consisted of context aware HMMs trained on 126-dimensional feature vectors.*

The results for the other more complex recognition tasks, such as random sentences, grammar based utterances, continuous speech were, as expected, less impressive. However, they are very promising because they are great improvements over the previous results. For instance in the case of the GU recognition task the best result obtained was WRR 48.36% and Acc 10.33%, while for the ALL recognition task the best result obtained was as small as WRR 15.94% and Acc -8.96%. Due to the difference in the complexity of the recognition tasks, a decrease in performance was expected. However, especially in the latter case we investigated more in depth to find the exact reasons. This was also because the results show a great difference between the word recognition rate values and the accuracy values. This difference appears because of a very high insertion rate. This is a big problem in the lip reading domain because anytime the mouth is moving the systems may interpret as speaking. This is not the case for speech recognition when the silence is very well defined. Therefore, a better definition of the silence models is needed. The insertion error is, however, not the only question to ask here. When we looked at the detailed results, we found that substitution error level was even higher. For instance for the GU task reported above the substitution rate was 39.93% while the insertion level was 38.04% and for the ALL task the substitution rate was 74.02% while the insertion rate was 24.90%. This picture shows that the systems are not very well trained in general, therefore, showing the amount of data samples in the training set is well under the optimum limit. However, in order to further investigate this, we build recognition systems for which we removed the language modelling layer which means that the results are obtained in term of viseme strings rather than in terms of word strings. For instance in the case of the GU task the results were WRR 56.89% and Acc 15.30%, substitution rate 25.07% and insertion rate 41.59%. There are two things to remark here. Firstly, we should remark that the recognition performance has increased. Secondly, and more importantly we should note here that while the substitution has decreased the insertion rate has increased considerably. These results explain the low recognition task but it does so in agreement with our expectance that the most errors are made in the non speaking part of the utterances. The increase in performance was observed in the case of the ALL recognition task as well: WRR 39.23% and Acc: 20.56%.

In order to get a better idea about the recognition results we investigated the confusion matrices. It is possible to visualize the viseme confusion matrix in all cases when the results are given in terms of viseme strings. However, when the results are given in terms of word strings, visualising the confusion matrix is feasible only in the case of digit strings and letter strings. In Figure 5.11 the images a) and b) show the confusion matrix obtained by the most successful recognisers, respectively. In the case of digit recognition the confusion matrix is almost perfectly diagonal. For the alphabet letters this is not the case anymore. We can notice that the most confused digit is the digit $< 1 >$ ([iee gkx]) and is often confused with digit $< 9 >$([gkx iee gkx at]). On the other side the letter $< A >$ ([aa]) is confused with $< H >$ ([h aa]), the letter $< C >$([sz iee]) is confused with $< D >$ ([td iee]), the letter $< G >$([gkx iee]) with $< D >$([td iee]), the letter $< N >$([eeh gkx]) with $< L >$([eeh l]), etc. Exactly the same pattern appears irrespectively of the number of mixtures we used. The images c) and d) from the same figure show the mean confusion matrices for each case. It can be seen that when a large part of the word transcription is similar, the confusion increases. It should be noted that the confusion matrix only shows the substitutions and in some small extent gives some insights about the deletions. The confusion matrix has some empty rows and columns. These elements correspond to the letters that due to the viseme definition have similar transcription in the viseme space. These pairs were listed in Section 2.5.3.



(a) **(b)**

(c) **(d)**

Figure 5.11: *The confusion matrices obtained by the best systems in the CD and CL tasks, respectively. a) the confusion matrix for CD task in the best case. b) the confusion matrix for CL task in the best case. c) the mean, over the mixture number, confusion matrix for the CD task. d) the mean, over the mixture number, confusion matrix for the CL task.*

We also investigated the results in an N-Best approach where the first 5 possible outcomes were considered. This approach gives the possibility to post process the results in order to choose the most probable outcome. However, the increase in the best result was rather marginal.

Even though it is not a common practice in speech recognition, but because in our case we still have a relatively small corpus compared with a regular corpus used for speech recognition, we analysed the results in a 10-fold validation experiment. In a 10-fold validation experiment the data is divided into 10 folders and each time 9 folders are used for training and 1 for testing. This means that we trained on 90% of the data and tested on 10% of the data in the corpus. This approach is meant to increase the certainty in the results, namely, one expects to have a low variance in the results obtained on different data. This is exactly what we found in our experiments. For instance in the case of the CD task we found the mean word recognition rate over the 10 folds was 88.82% with a standard deviation of 1.48%.

The maximum word recognition rate was 91.39%. Similarly, the mean accuracy rate found was closer to the one obtained in the first experiment, namely 74.50% with a standard deviation of 3.73%. The maximum accuracy obtained was 79.43%. In all the experiments we noticed a very large insertion rate which agrees with the previous findings. The best results are summarized in Table 5.1.

Table 5.1: *The best results obtained based on the SLGE feature extraction method.*

Task	WRR	Acc
CD	91.39%	79.43%
CL	56.34%	17.16%
GU	56.89%	15.30%
All	39.23%	20.56%

5.3 Conclusions

In this chapter we investigated the use of a statistical approach for estimating the shape of the lips for lip reading. We presented two methods for improving the process of lip geometry estimation, namely, region of interest detection and outlier detection. These improvements make the method more robust to the changes in performance of the colour filter used. By using an object detection algorithm to find an accurate bounding rectangle around the mouth we remove much of the face areas on which the colour filter is prone to make errors. Even further, the outlier detection approach produces smoother and more accurate mouths shapes. From the point of view of the processing complexity, we found that this approach is very fast, and could be successfully deployed in real time applications. However, we should stress here that the development of a suitable colour space and accompanying filter would make this method more universally applicable.

The results obtained based on this approach show great improvements over the previous results. In the case of the simpler tasks, like digit strings and letter string recognition we achieved results comparable with the previous results. However, in the case of the more complex systems we found great improvements. This result is a clear indication that this method can be successful for more complex systems. However, in order to achieve good performance we need sufficiently more data and of better quality (i.e. we have shown in the previous chapters that the corpus we used was specially built for the current research, and that it is considerably larger than the previous data corpus).

Chapter 6

Active Appearance Models for Lip Reading

Active Appearance Models(AAM) is a model based approach for image segmentation. The definition and the theoretical aspects of this approach were introduced in Section 3.4 of Chapter 3. In the current chapter we describe the use of AAM for lip reading. We first define the set of visual features used and, thereafter, we present and analyse the obtained results.

6.1 Description of the Algorithm

The algorithm starts by applying the AAM searching scheme in order to detect the shape of the mouth. Based on the locations of the points that define the mouth shape we compute a set of geometric feature that are used to train the recognition systems.

6.1.1 Facial Model for Lip Reading

As shown in Section 3.4, the AAM algorithm iteratively searches for the best fit of a model defined by a set of landmarks and the image being processed. Based on a-priori knowledge about the shape of the object, the set of landmarks is defined such that it optimally describes the object. In our case we required that the points selected describe the shape of the mouth in detail, especially capturing the speech related aspects. Therefore, the final model should exactly segment the lips in all moments during speech. After experimenting with different models and analyzing the results, followed by long discussions, we decided to use a model composed of 29 points, distributed around the mouth, chin and nose. This model is shown in Figure 6.1.

For training a model, a number of two to four hundred images was manually processed. In order to obtain reliable results the images were selected such that

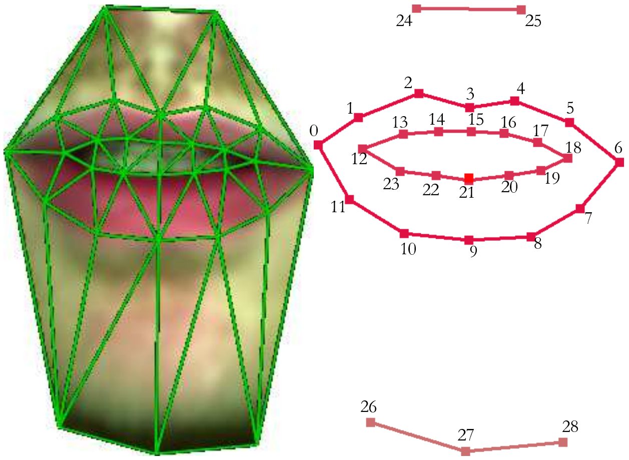


Figure 6.1: *The AAM model.*

they cover all the variance in the data. This was achieved in an iterative process. We first started with a random selection of a few tens of images which were used to build a first model. This model was used for processing until the performance of the model decreased below some visually assessed threshold. The images that were badly processed were added in the training set and a new model was obtained. This process continued until the performance of the model stabilized. In the end we trained a number of models for each speaker in the dataset. For speakers that recorded multiple sessions we trained one model per session. Even though the process is fairly automated, this was an extremely laborious work, since the corpus contains more than 4.3 million frames, and was split among various people. Each assistant was asked to train a model and supervise the processing of the rest of the frames. Splitting the data among different people makes it more difficult to guaranty the uniformity over the entire corpus of the end result. Therefore, to assure uniformity of the processing we used a strict definition of the landmarks. We defined as well constraints that acted on pairs of landmarks. The rest of this section gives the definition used for the landmarks. Before going to the next paragraphs, we should introduce some anatomical elements on which the definition of the landmarks depends. Figure 6.2 shows the anatomy of the mouth area.

Outer Mouth Contour The points on the outer mouth contour are defined as follows (see Figure 6.1):

1. Point 0 is the leftmost point of the lips (i.e. left mouth corner).

2. Point 6 is the rightmost point of the lips (i.e. right mouth corner).

3. Points 2, 3 and 4 are placed in accordance with the philtrum (infra nasal depression), namely, 2 and 4 at the foot of the philtral columns, respectively,

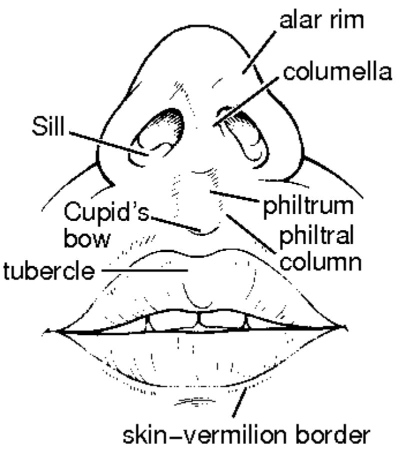


Figure 6.2: *The mouth area anatomy.*

and 3 in the place where the philtrum meets the upper lip, also called the philtral dimple.

4. Points 8, 9 and 10 correspond to points 4, 3 and 2, respectively.

5. Points 1, 5, 7 and 11 are placed such that the lip area is covered as closely as possible, and should lie on the skin-vermilion border. However, their positions are preferred to be at equal distances from their neighbouring points.

It should be noted that the outer mouth contour contains much person-dependent information.

Inner Mouth Contour In the case of the inner mouth contour the decision was that the stress should be placed on accurately describing the aperture of the mouth. A closed mouth is a special case. The points on the inner contour are closely related to the ones on the outer contour and have similar definitions.

1. Point 12 is the leftmost point of the cavity of the mouth. However, in the case of a closed mouth this is not possible to observe. In that case this point should be placed such that it best describes the mouth line, but always to the left of points 13 and 23.

2. Point 18 is the rightmost point of cavity of the mouth. In a similar way as for point 12, in case of a closed mouth this point should be placed such that it best describes the mouth line, but always to the right of points 17 and 19.

3. Points 15 and 21 correspond to points 3 and 9 and follow the philtrum. An additional constraint was that the points 3, 9, 15 and 20 should always belong to the same line.

4. The last 8 points form pairs as follows: 13 and 23, 14 and 22, 16 and 20 and 17 and 19, and have similar definitions as the points 1, 5, 7 and 11, namely they should divide the distance between the adjacent points in three equal segments (e.g. for instance the points 16 and 17 divide the lip contour between the points 15 and 18 in three equal segments).

Nose The nose points are only a delimitation of the nose and are used as a reference to compute distances to and between other points (e.g. the distance from point 27 on the chin to the line formed by points 24 and 25 is used as a feature in our settings). The points are placed at the base of the nose. This distance was also the base distance that was used to normalize all other distances on the speaker's face. We used this distance because we found out that through careful annotation the standard deviation of this distance during an entire recording was always less than 2 pixels.

Chin Here we are interested in tracking the tip of the chin marked by point 27. Points 26 and 28 only support the detection of point 27 and should be placed symmetrical with respect to the point 27 and describe the chin as closely as possible. Point 27 should be aligned to points 3, 9, 15 and 21.

6.1.2 AAM Results on the Training Data

The AAM process is very fast and very accurate given that a good training set was selected. We combined the AAM searching scheme with the Viola/Jones mouth detection algorithm, which made the selection of a very good location for the initial guess possible. This has speeded up the search process to real time performance. The mouth detection was used only in the first few frames of the recording. In the subsequent frames the initial guess used was the result of the processing in the previous frame. This approach was very successful both in speeding up the search scheme and improving the accuracy of the detection.

Figure 6.3 shows the first six most important components in PCA terminology. The mean shape and texture model is shown on the centre row. The top row shows for each mode the resulting object after an adjustment by two standard deviations is applied to on the corresponding mode. The bottom row shows the result when the adjustment is negative. The first two modes seem to have more control over the vertical and horizontal movement of the mouth, while mode four seems to control the presence of the tongue. However, there is no strict separation between the information controlled by each mode, at least not easily discernable by visual inspection. This model was trained on a set of 440 images, selected in an iterative process. All three models (i.e. appearance, shape and combined models) were truncated at 95% level. Based on the 95% level truncation, as can be seen in Tables 6.1, 6.2 and 6.3, the final combined model had 38 parameters, while the shape model had 11 parameters and the texture model had 120 parameters. These tables also show the way each new parameter covers the variance in the training data. The first six modes in the combined model cover 78.65%. However, in the case of the shape models the first two modes already cover 82.53% of the total variation, while the first six cover 91.83% of the variation.

Table 6.1: *Combined AAM model, mode variation.*

	per mode	cumulative		per mode	cumulative
1	35.07%	35.07%	20	0.42%	90.75%
2	25.69%	60.76%	21	0.39%	91.14%
3	7.39%	68.15%	22	0.36%	91.50%
4	4.85%	73.00%	23	0.33%	91.83%
5	3.28%	76.28%	24	0.30%	92.13%
6	2.36%	78.65%	25	0.29%	92.42%
7	1.80%	80.45%	26	0.27%	92.69%
8	1.34%	81.79%	27	0.25%	92.94%
9	1.29%	83.08%	28	0.24%	93.18%
10	1.28%	84.35%	29	0.23%	93.40%
11	1.04%	85.39%	30	0.22%	93.62%
12	0.82%	86.22%	31	0.21%	93.83%
13	0.73%	86.95%	32	0.20%	94.02%
14	0.72%	87.67%	33	0.19%	94.21%
15	0.67%	88.34%	34	0.18%	94.39%
16	0.57%	88.91%	35	0.17%	94.56%
17	0.50%	89.41%	36	0.16%	94.72%
18	0.49%	89.90%	37	0.16%	94.88%
19	0.43%	90.33%	38	0.15%	95.03%

6.1.3 Defining the Feature Vectors

The first approach towards lip reading and other similar problems was to use as visual features directly the AAM parameters as defined by equation 3.75. The other approach is to use the final results of the method, namely the co-ordinates of the landmarks as assigned by the algorithm for the current image. In our research we adopted this latter approach. Based on the position of the landmarks we defined seven high level geometric features.

The features are computed as the Euclidean distances and areas between the certain key points that describe the shape of the mouth, namely mouth height and width, mouth aperture width and height, mouth area, aperture area and the nose to chin distance. In terms of the landmarks locations the features are defined based of the points as shown in Figure 6.1 as follows:

1. Mouth height is defined as the Euclidian distance between points 3 and 9.

2. Mouth width is defined as the Euclidian distance between points 0 and 6.

3. Mouth area is defined as the area inside of the outer lip contour.

Table 6.2: *Shape based model, mode variation.*

	per mode	cumulative
1	51.96%	51.96%
2	30.58%	82.53%
3	3.32%	85.85%
4	2.89%	88.75%
5	1.82%	90.57%
6	1.26%	91.83%
7	1.01%	92.84%
8	0.83%	93.67%
9	0.80%	94.46%
10	0.52%	94.98%
11	0.39%	95.37%

4. Aperture height is defined as the largest Euclidian distance between the pairs of points (13, 23), (14, 22), (16, 20), (17, 19) and (15, 21).

5. Aperture width is defined as the Euclidian distance between the leftmost point (or coinciding pair of points) and the rightmost point (or coinciding pair of points) of the inner lip contour.

6. Aperture area is the area covered by the mouth aperture, namely the inner lip contour.

7. Nose to chin distance is the minimum distance between chin point 27 and the line defined by the nose points. This denotes the openness of the jaw.

The features are graphically described in Figure 6.4.

6.1.4 Visual Validation of the Feature Vectors

Figure 6.5 shows the plots of the feature vectors computed for a random recording of the letter F having the viseme transcription [eeh fvw]. In this case the onset and offset moments of the utterance are clearly visible around the frame 75 and the frame 200 of the video recording, respectively. The onset of the viseme [eeh] is around the frame 80, while the onset of the viseme [fvw] is seen around frame 160. The actual shape of the mouth can be seen in the images shown below the graphs, which are extracted from the video sequence.

Table 6.3: *Appearance based model, mode variation.*

	per mode	cumulative		per mode	cumulative
1	29.16%	29.16%	20	0.55%	80.72%
2	13.30%	42.47%	21	0.53%	81.26%
3	8.39%	50.86%	22	0.50%	81.76%
4	6.97%	57.82%	23	0.46%	82.22%
5	4.25%	62.08%	24	0.44%	82.66%
6	3.12%	65.20%	25	0.41%	83.07%
7	2.43%	67.62%	26	0.39%	83.45%
8	1.90%	69.53%	27	0.39%	83.84%
9	1.62%	71.14%	28	0.36%	84.20%
10	1.49%	72.64%	29	0.33%	84.53%
11	1.30%	73.94%	30	0.32%	84.86%
12	1.04%	74.98%	31	0.31%	85.17%
13	0.94%	75.92%	32	0.29%	85.46%
14	0.89%	76.81%	33	0.28%	85.74%
15	0.81%	77.62%	⋮	⋮	⋮
16	0.72%	78.34%	117	0.05%	94.88%
17	0.65%	78.99%	118	0.05%	94.93%
18	0.60%	79.59%	119	0.05%	94.97%
19	0.58%	80.17%	120	0.05%	95.02%

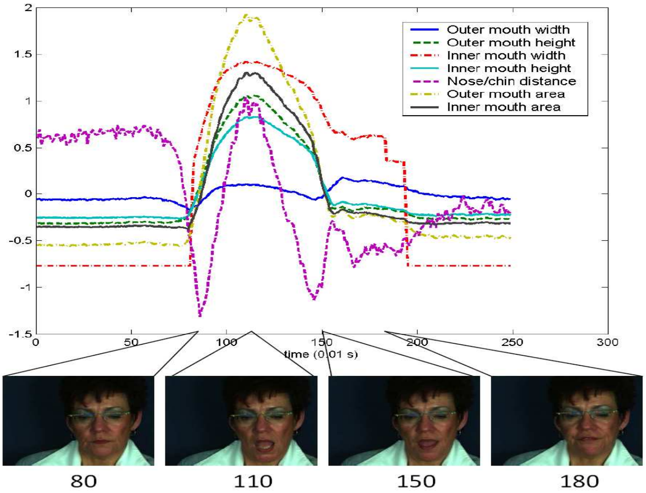


Figure 6.5: *The seven features plotted for one recording for the letter F transcribed using the visemes: eeh and fvw.*

Figure 6.6 shows the plots of the feature vectors for seven letters of the alphabet

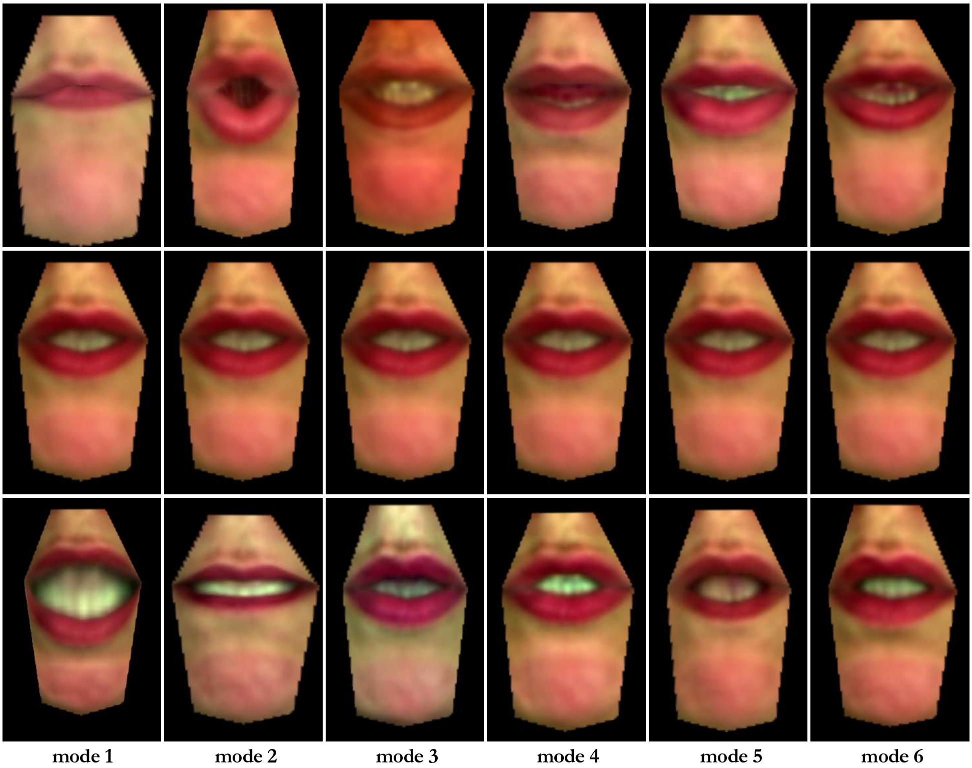


Figure 6.3: *Combined shape and appearance statistical model. The images show from left to right the first six most important components in PCA terminology. These modes account for 78.65% of the total variation. Centre row: Mean shape and appearance. Top row: Mean shape and appearance $+2\sigma$. Bottom row: Mean shape and appearance -2σ.*

and the digit $< 8 >$ ([a gkx td]). We see that the variability of the features is very high which makes them suitable for the recognition task at hand. We can also remark that, for instance, even though the viseme [aa] is present in the transcription of all letters, A([aa]), H([h aa]) and K([gkx aa]) we can clearly see that there is a slight difference between them with respect to the duration in each instance. This is best visible in the curve showing the height of the mouth, which shows that the duration of the viseme is shorter in the utterance of the letter K and H than in the case of the letter A.

An interesting result was obtained when visually inspecting the curves described by the feature vectors for all the visemes. By simple visual inspection we found that we could easily distinguish between some of the visemes, which proved that the feature set captures much of the speech related information. Table 6.4 summarises our findings in this respect. For a simple recognition task such as for instance the recognition of isolated visemes, or even the recognition of isolated digits, based on this table we could use a static classifier such as Support Vector Machines(SVM) [Gan02]. However, for these types of classifiers the features need to be global features

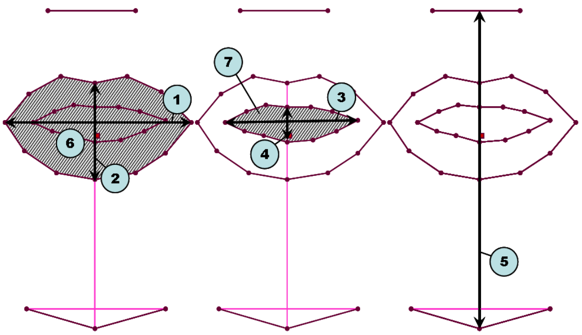


Figure 6.4: *The high level geometric features: 1) Outer lip width, 2) Outer lip height, 3) Inner lip width, 4) Inner lip height; 5) Chin to nose distance, 6) Outer lip area, 7) Inner lip area.*

Table 6.4: *Feature patterns per viseme: +) peak -) valley -+) increase +-) decrease.*

	aa	h	gkx	a	oyu	ie	ei	iee	td	sz	eeh	l	pbm	fvw	at
Outer width	-		+	-	-	+	-+	+	+	+	+-	+-		+-	
Inner width	+		+	+		+	+	+			+		-		
Nose/chin dist	-	+	+	-	-	-	-	+-	+		+-	+-			
Height/area	+	+	+	+	-	+	+	+			+				

because they cannot handle time series. Therefore, the generalisation to longer and of variable length utterances is not possible.

6.2 AAM as ROI Detection Algorithm

It is worth mentioning that AAM can be used as well as a preprocess for defining a more accurate ROI. Therefore, the ROI defined using a mouth detection algorithm is further improved using the AAM. A more accurate ROI makes the data parametrization process more robust, because the background is better removed and, therefore, there is less noise in the input data. This can be used in both the other data parametrization methods introduced in this thesis, and we actually used it for the optical flow approach introduced in Chapter 7. We can also say that this method would have a similar result to the shape estimation introduced in Chapter 5. The colour filtering would be trivial if the ROI were defined using the AAM approach and the pixels distributions would be exactly reflecting the real shape of the mouth.

6.3 Lip Reading Results

The method presented in this chapter produces for each frame in the corpus a vector with seven entries: mouth width, mouth height, aperture width, aperture height, mouth area, aperture area and the distance between the nose and the chin. We trained and tuned a lip reader based on the HMMs approach for each recognition task. In a similar approach as described in Chapter 5, we considered both the case with simple static features (i.e. seven geometric features) and the case when the feature space was enriched with dynamic information consisting of deltas and accelerations (i.e. making 21-dimensional vectors). We trained systems based on mono-visemes as well as context aware tri-viseme systems. We used a Gaussian mixture arrangement to better describe the feature space and we performed a 10-fold validation in order to increase the confidence in the observed results. The best results obtained were WRR 90.32% with word accuracy 84.27% for the CD recognition task. In this case, 75% of the sequences was recognized correctly. Figure 6.7 shows the plot of the performance of the best recognizer as a function of the number of Gaussian mixtures used.

For the GU recognition task we observed a 56% WRR. Using an N-Best approach with five most probable outcomes did not improve the result, which suggests the system is fairly robust. The 10-fold validation showed an 80.27% mean WRR with a 6% standard deviation, the minimum performance being 74.80% WRR. This shows some instability, however, the minimum is still a very good result. We also tested the results of the recognition at viseme level (i.e. before using the language model to build the corresponding words). This is useful for analysing the degree of confusion between different visemes. Figure 6.8 shows the confusion matrix for the best case. The mean confusion matrices computed over the mixture number is also displayed. We can remark in these figures that the degree of confusion is relatively small. However, the confusion is greater for visemes defined by larger phoneme sets. This is the case especially for the visemes [oyu] and [gkx] which are very often a source of confusion.

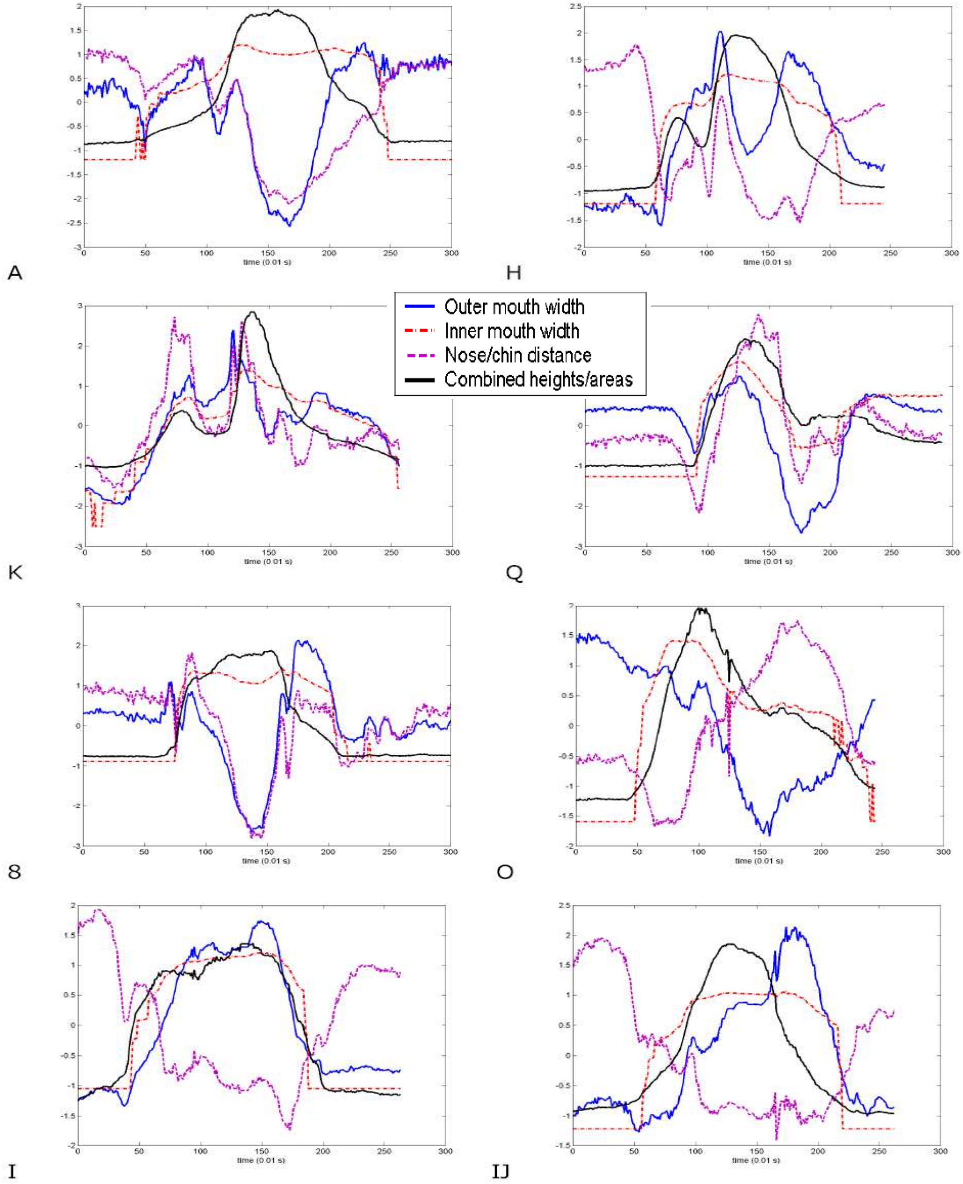


Figure 6.6: *Feature values plotted for the letters A ([aa]), H ([h aa]), K ([gkx aa]) and Q ([gkx oyu]), I ([ie]), O ([oyu]), IJ ([ei]) and 8 ([a gkx td]). The vectors are scaled using the time variance and centred around their mean.*

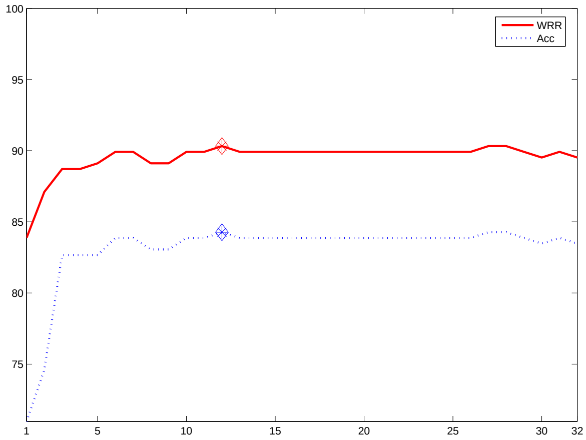
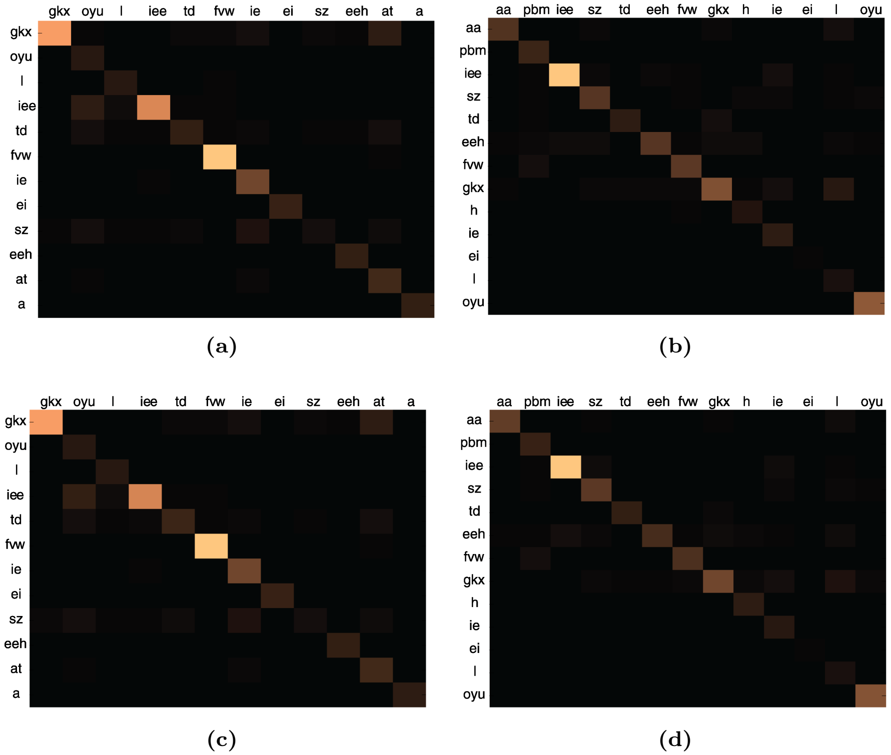


Figure 6.7: *The WRR and Acc results for CD recognition task as a function of the number of mixtures. The x axis gives the number of mixtures and the y axis shows the results obtained. The feature vectors consisted of geometric features computed based on the AAM shape corroborated with their corresponding deltas and accelerations. The HMM models consisted of intra-word tri-visemes.*

6.4 Conclusions

We introduced in this chapter an AAM based approach for lip reading. The AAM method is in our opinion a valuable tool for lip reading, both as a data parametrization method but also as a ROI detection technique. The method can be very robust and has a good generalization for unseen faces, however, the training process can be very long for satisfactory results to be obtained. Nevertheless, the shape obtained from the search scheme can be used as a starting point for testing other feature types, since it can always function as background elimination stencil.

Based on the shape computed using the AAM searching scheme, we defined a set of high level geometric features. Based on these features we built different lip readers with very good results. These results validate the findings reported in the literature which showed that the width and the height of the mouth largely capture the content of the spoken utterance [Woj03]. This also justifies why a simple mouth model for lips synchronization based only on varying the mouth opening synchronous with the sound output is so convincing. We did not include in the feature vectors used in this chapter any information that describes the presence of the teeth, tongue or other elements of the mouth. This information was shown in the literature but also in our other experiments to be very important for lip reading. We expect that this is the case in the current settings as well. However, we did not include this information here because we wanted to have a clear understanding of the factors that influence the observed results.



(a) **(b)**

(c) **(d)**

Figure 6.8: *The confusion matrices obtained by the best systems in the CD and CL tasks at the viseme level, respectively. a) the confusion matrix for CD task in the best case. b) the confusion matrix for CL task in the best case. c) the mean, over the mixture number, confusion matrix for the CD task. d) the mean, over the mixture number, confusion matrix for the CL task.*

Chapter 7

Optical Flow Analysis for Lip Reading

Optical flow represents the apparent movement in a sequence of images and, therefore, links together consecutive video frames. Computing the optical flow around the speaker's mouth represents an alternative for the static analysis of the input video, because it directly describes the actual movement of the mouth. The use of optical flow analysis for lip reading goes back to the beginning of this domain. However, the computational complexity of detecting the optical flow made its use very restrictive. Even with the development of better algorithms and more powerful CPUs it is still one of the slowest approaches for data parametrization for lip reading. Recently, successful experiments were made to implement the optical flow algorithms directly into hardware, which provides real time capabilities [Cor02; Mar05; Día06]. Optical flow is an extremely valuable technique for motion detection and obstacle avoidance with applicability in automatic navigation. This is a defining necessity for Unmanned Vehicles (UV) but also a source of increased security for regular vehicles, therefore is a very attractive feature for the automotive industry.

In the previous approaches the optical flow was used either as raw data for the lip reader or as a measure of the overall motion on the speaker's face and, therefore, as a method for detection the onset/offset moments. In this chapter we present a set of features defined starting from the optical flow field computed in a carefully chosen area around the speaker's mouth and analyse the performance of the lip readers built based on these features.

7.1 Description of the Algorithm

The first step made is to carefully define the region of interest, where the optical flow will be computed. This is necessary both for increasing the accuracy of the algorithm but also for speeding up the detection of the optical flow. The optical flow is computed inside the area of interest. We used both the algorithm developed by Lucas and Kanade and the algorithm developed by Horn and Schunk but we obtained more reliable results with the latter algorithm. However, any fast and accurate optical flow algorithm can be used. From the computed optical flow field we define a set of features that describe the amount of movement around the mouth. These features are then used to train the recognition models. Figures 7.2 and 7.1 show two examples of the optical flow computed for some sequences of frames during closing and, respectively, opening of the mouth.

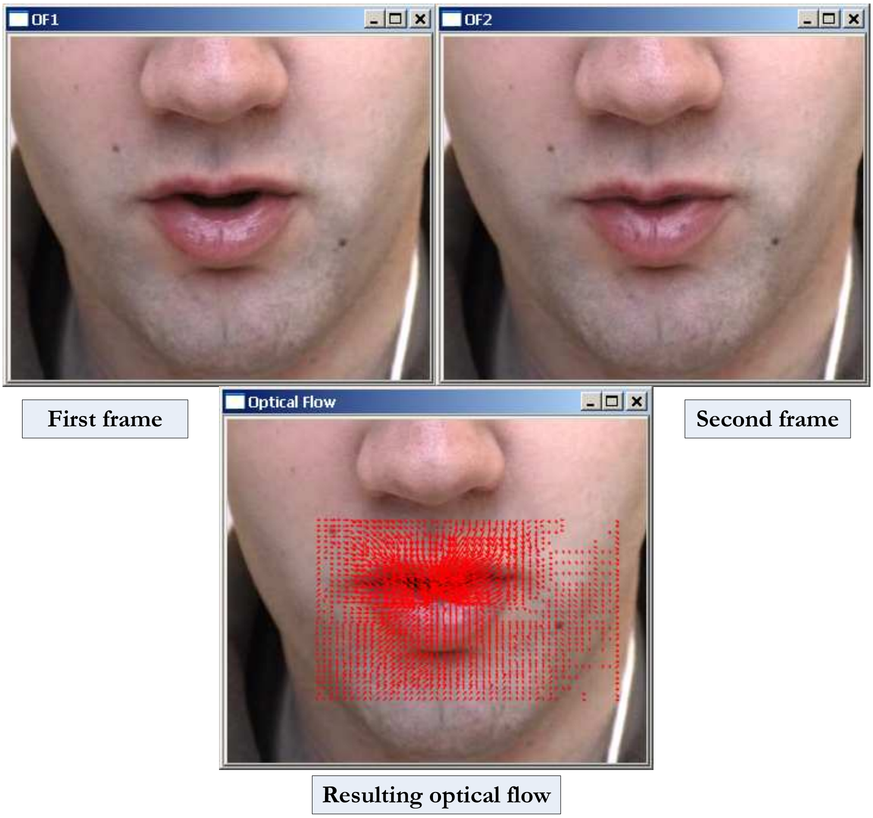


Figure 7.1: *Example of the optical flow computed during closing of the mouth.*

7.1.1 Defining the Region Of Interest

In this case it is very important to define a strict boundary of the space where the optical flow will be computed. As in the other cases we used an object detection algorithm to compute a bounding box around the mouth. Subsequently this bounding box was further processed using the AAM searching scheme. The results of the AAM process was also used for normalizing the input frames such that the centre of the mouth in all the frames was translated to the same virtual location and the images were scaled so that to have the same dynamic range. This process removed the influence of the camera location on the resulting images, making a uniform processing of the input images from different recording sessions possible. In the case

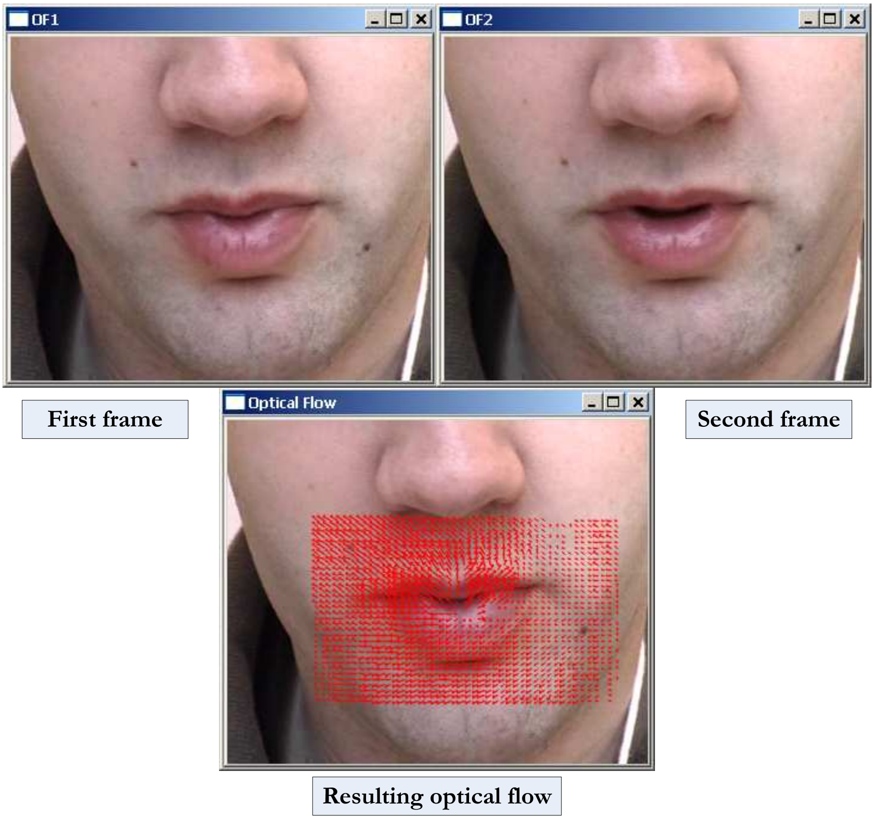


Figure 7.2: *Example of the optical flow computed during opening of the mouth.*

of the optical flow analysis this is necessary because we need to be sure that the observed flow corresponds to the same real locations in all the images.

7.1.2 Defining the Feature Vectors

Instead of using the optical flow directly as raw data we want to have a smaller set of features that accurately describes the movement around the contour of the mouth. Therefore, the method defines a set of regions around the mouth from which some global measures of the optical flow will be used as visual features.

The first step of the method is to define the set of regions. We used a radial arrangement of the regions centred in the centre of the mouth, even though other arrangements are also possible. For instance, for facial expression recognition in [Sun09] the authors define 15 rectangular areas located in specific areas on the face, in order to measure the different facial muscle activation. As in the case of the algorithm presented in Chapter 5 and shown in Figure 5.3, the face was divided into 18 equally wide sectors centred in the centre of the mouth. The centre of the mouth can be computed either based on the AAM solution or based on the estimation technique introduced in Chapter 5, however, due to its accuracy we believe that the AAM solution gives more reliable results.

The visual features are extracted from each of the identified sectors and should describe the amount and the direction of the motion. Therefore, each feature is obtained by computing statistics of the optical flow in all 18 sectors directions, respectively. Even though the global variance of the optical flow can be valuable information for onset/offset discrimination, the information about movement in certain parts of the mouth is cancelled out by averaging, hence is not suitable for lip reading. This is because for a small enough angle the contour of the lip will deform in the same way on the entire distance considered. Because of this, the variance of the flow computed in one sector is expected to be always zero and to hold no information about the spoken utterance. The mean displacement on the other side will show both the magnitude of the movement and also the direction of the movement in any given radial sector. We computed, therefore, as visual feature only the mean horizontal and vertical displacements, respectively. Figure 7.3 shows the features extracted based on the optical flow seen in Figure 7.2.

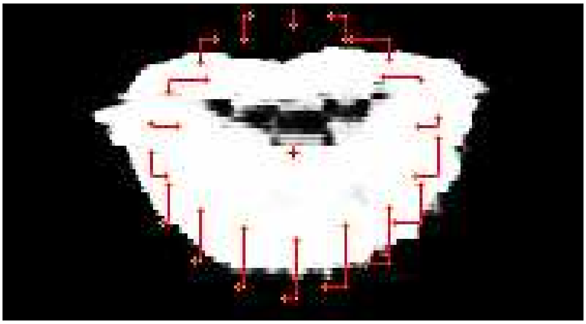
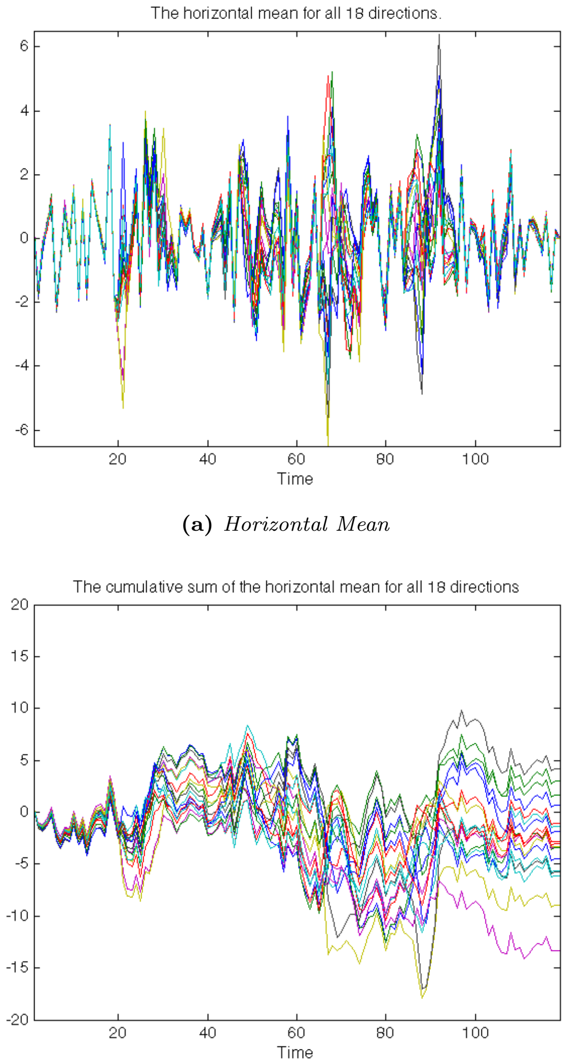


Figure 7.3: *Optical flow based extracted features.*

7.1.3 Visual Validation of the Feature Vectors

Figures 7.4 and 7.5 show the plots in time of the features computed for a random utterance in the corpus. In these figures the second image shows the same features but cumulated over time. The values for all 18 sectors are depicted. The utterance spreads over 119 video frames. While it is clear from the images of the actual features that the mean displacements in optical flow carry a significant amount of information, there is no high level information that we could immediately spot from these images. However, when looking at the graphs showing cumulative quantities, we immediately see a pattern which has meaningful information about the processed utterance. This is the case especially in graph b) in Figure 7.5, where the mouth movement can be easily followed.



The horizontal mean for all 18 directions.

(a) *Horizontal Mean*

The cumulative sum of the horizontal mean for all 18 directions

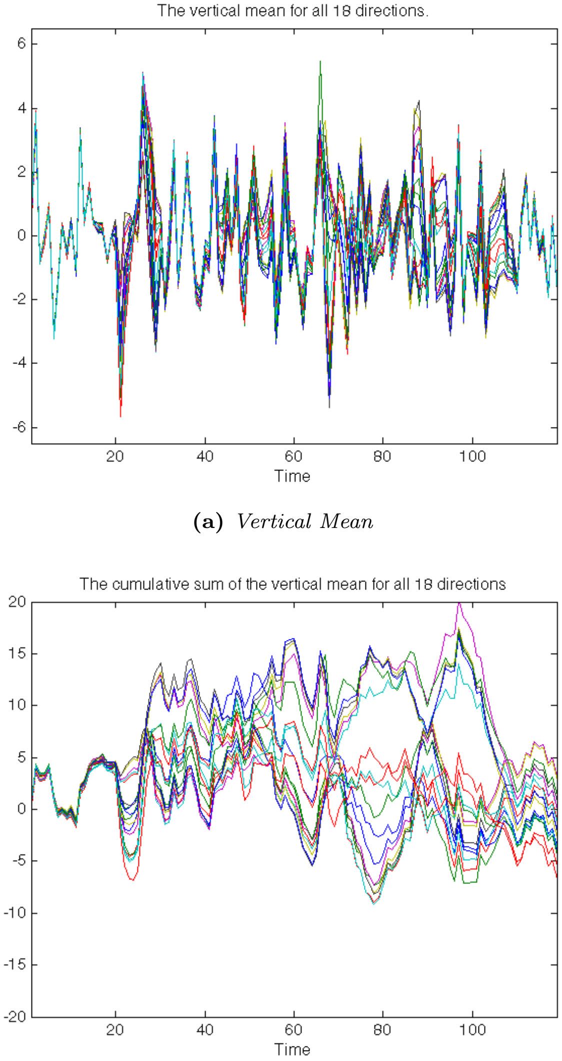
(b) *Cumulative Horizontal Mean*

Figure 7.4: *The distribution over time of the horizontal mean of the displacement in the optical flow for a random utterance. Each line corresponds to one of the 18 distinct directions: a) the actual values; b) the cumulative sum.*

7.2 Lip Reading Results

In this case for each video recording a 36 dimensional feature vector, (V_x, V_y), was computed for all the pairs of two consecutive frames. We trained as in the other case many different recognisers depending of the particular settings.

However, when the time came to test the trained systems, the results we observed



(a) *Vertical Mean*

(b) *Cumulative Vertical Mean*

Figure 7.5: *The distribution over time of the vertical displacement mean in the optical flow for a random utterance. Each line corresponds to one of the 18 distinct directions: a) the actual values; b) the cumulative sum.*

were less impressive than what we were hoping for. Of course, the worst fact was that we could not understand why the performance was less optimal, since we were using high speed good quality recordings and we carefully processed the data in the same way we did in the other cases. The best result we obtained was WRR +60.29%, but the accuracy was as low as +0.00%. To understand the motives for these less optimal figures we split the results into their constituent parts. We found that there were 29.66% words substituted, 10.05% words deleted and an impressive 60.29% words inserted. The latter quantity is responsive for the very low accuracy. To be sure that these results are reliable, we ran a 10-fold validation experiment. We found that the results are consistent over the different folds. The insertion was even found to go to a maximum of 77.02%. Table 7.1 gives the results for the same task recognition over the 10-folds.

Table 7.1: *10-fold validation experiment with optical flow based features.*

WRR	Acc	Insertion
+50.19%	+18.92%	+31.27%
+53.63%	-23.39%	+77.02%
+55.19%	+13.11%	+42.08%
+52.25%	+8.30%	+43.94%
+58.28%	+28.22%	+30.06%
+52.83%	+22.17%	+30.66%
+58.06%	+31.45%	+26.61%
+60.29%	+0.00%	+60.29%
+51.54%	-28.19%	+79.74%
+51.60%	+4.40%	+47.20%
Mean values		
+54.39%	+7.50%	+46.89%
Std values		
+3.42%	+20.21%	+19.44%

In order to better understand the causes behind this situation, we investigated the results at the level of visemes. Table 7.2 gives the results at viseme level. As seen in this table, the recognition in terms of WRR is at least 20% higher than the previous results (i.e. this is usually expected since these are the recognition results before using the language model to refine them. The language model is supposed to bring the extra information needed.) However, due to an enormous insertion rate, the accuracy touches extremely low limits. The obvious conclusion was that there is definitely something sub-optimal going on.

Before presenting the results of our investigation it is worth commenting here on another interesting finding. It turned out during the analysis of the results that in this case the addition of the deltas and accelerations to the features vectors had a marginal effect (i.e. maximum 5% increase) on the performance of the system. This was actually expected because the optical flow features already contain dynamic information, and therefore, the first derivatives of the data features do not bring much new information.

The only explanation to this state of facts was that the computed optical flow was not accurate enough. We found out that this was actually the case. Because of the high speed recording, the difference between two consecutive frames was so small that it would usually go beyond the accuracy of the optical flow detection procedure. During the times with little facial muscle activity, the optical flow algorithm had rather random results which would register as noise for the recognition engine.

Table 7.2: *10-fold validation experiment with optical flow based features. The results are given a viseme level.*

WRR	Acc	Insertion
+74.66%	-158.93%	+233.58%
+74.28%	-153.79%	+228.07%
+75.66%	-164.02%	+239.68%
+73.92%	-195.46%	+269.39%
+74.29%	-141.09%	+215.38%
+73.44%	-139.39%	+212.82%
+74.20%	-140.56%	+214.76%
+73.89%	-180.65%	+254.53%
+76.24%	-166.29%	+242.53%
+74.68%	-157.62%	+232.30%
Mean values		
+74.53%	-159.78%	+234.30
Std values		
+0.85%	+18.09%	+18.20

However, the highly dynamic muscle activity was always captured, therefore making the correctly recognized viseme rate very high. In order to test this theory, we down-sampled the video stream to a $25Hz$ rate by removing frames from the recording. We re-computed the optical flow for the new visual data stream. We computed the new visual features based on the newly computed optical flow field and we trained the lip reader system again. The results were entirely as expected. The insertion rate greatly decreased and both the word correct rate and the accuracy increased considerably, especially the accuracy. Therefore, the WRR found now was 70.66% and the accuracy was 65.64%. However, the performance at the level of visemes decreased slightly to +66.42% WRR. The accuracy found in this case was +40.07%. If there is still a lag in performance, this is most probably due to the imperfections of the optical flow detection algorithm. In order to test this possibility we would need to compute the ground truth of the optical flow. However, in our settings the ground truth is very difficult to obtain.

7.3 Conclusions

In this chapter we introduced a set of features computed based on the optical flow computed between consecutive frames in the input video. We presented and analysed the resulting performance of the lip reading systems when the training was done based on the new feature frames. We found out that due to the performance of the optical flow algorithms the use of high speed recordings was actually more disturbing than promoting good recognition results. This is because when there is low activity facial motion the optical flow algorithms behaves rather randomly increasing the noise in the system. However, in the case of standard speed recordings the results show healthy recovery to normal levels. There is, therefore, a conflict between this approach and the need for high speed recordings required by the speech rate variability.

Chapter 8

Further Analysis of the Results and Other Experiments

We analysed in the previous chapters the results of the lip readers trained based on different feature extraction approaches. We concluded that all methods made obtaining high recognition rates possible. Each method was presented separately and we emphasised the strengths and weaknesses for each approach while presenting the actions that were taken in order to improve the recognition.

In this chapter we compare the results obtained with respect to different aspects. Firstly, we compare the results obtained with respect to the data parametrization approach. Secondly, we analyse the results taking into account the video recording rate. Thirdly, we analyse the results from the point of view of the speech rate.

8.1 Comparison Based on the Feature Extraction Method

This section combines the results presented in the previous chapters into one single place, making it easier to compare the performance of the systems trained on different data parametrization approaches. We should make clear here, one more time, that the systems were trained using 100Hz video recordings. However, as shown in Chapter 7 for the optical flow approach using 100Hz video data was actually performing poorer than the 25Hz. Even in this case the optical flow approach produced slightly higher recognition faults. On the other side the AAM and SLGE based lip readers achieved good results. The best performances obtained were in both cases around 90% recognition levels. The 10-fold validation shows a slightly better performance by the SLGE system with a mean of 88.82% compared to 80.27% in the AAM case. However, in the AAM case the extra information describing the appearance of the cavity and teeth was not used, which can explain this slight difference. One interesting finding was that the two approaches did not performed entirely synchronized with respect to the folding. For instance the SLGE obtained its highest result in fold 8, namely WRR 91.39% and Acc 79.43%, while the AAM obtained the highest score in fold 7, namely WRR 90.32% and Acc 84.27%. SLGE obtained WRR 88.71% and Acc 70.56% in fold 7, while AAM obtained only WRR 78.47% and Acc 67.94% in fold 8.

More parallels between the different methods can the found in Section 8.3.

8.2 Comparison Based on the Type of Speech

The variance of the speech rate was the reason we decided to record at a high speed rate. This section investigates the influence of the speech style on the performance of the systems. All results shown here were obtained using systems with no fine tuning taken place. We took this approach in order to eliminate other influences on the results than the speech style. The results were, as expected, highly dependent on the speech style, namely the results for normal speech were in some cases a few times better than the results for high speech rate. For instance, in one experiment for GU utterances we observed a decrease from 35.48% in the case of normal speech rate to an extremely low 5.88% in the case of high speech rate. When considering the 5-letter word speech rate definition, the results were slightly better, but this is understandable since it generates a larger utterance set.

An interesting finding was that in many instances the utterances consisting of whispering and spelling in the same time were classified as high speech utterances and therefore were poorly recognised. For instance we observed a decrease from 29% to 12.50%. In case when the whispering was done at normal speech rates, the recognition performance achieved has approached, however from below, the recognition performance of the normal speech (e.g. for instance we found a decrease from 28.02% to 20.69% but also more or less similar values such as 25.41% and 25.00%). One remark is that in general the speech style distribution is not uniform in different recognition tasks. For instance the digit sets were usually spoken using normal speech, while the whispering was classified as having high speech rate. Therefore, the letter strings were classified in general as having normal speech rate. However, the letter strings which were produced as a result of spelling were in general classified as having high speech rate. The conclusion is that the letter string recordings contain higher speech rate than the digit string recordings.

Interestingly, the systems based on the AAM features performed better when the general settings were considered, namely when the systems were trained without including any fine tuning, such as Gaussian mixtures or intra-word context. For instance the AAM based system obtained 42.13% WRR on the normal speech subset, while the SLGE obtained only 30.27% on the same utterance set. This result remains valid for the high speech rate as well. However, as seen in the first section of this chapter, when fine tuning was used, both approaches achieved the same level.

Speech rate can also be analysed in terms of the correlation with the lips articulation. When the speech rate is increasing speakers tend to shorten the vowels, omit the endings of the words and link them together and more evidently to reduce the movement of their lips. Therefore, the variation in the lips movement decreases. Some speakers have a low lip movement even in normal situations. In this situation the gain obtained by increasing the recording rate is also diminished.

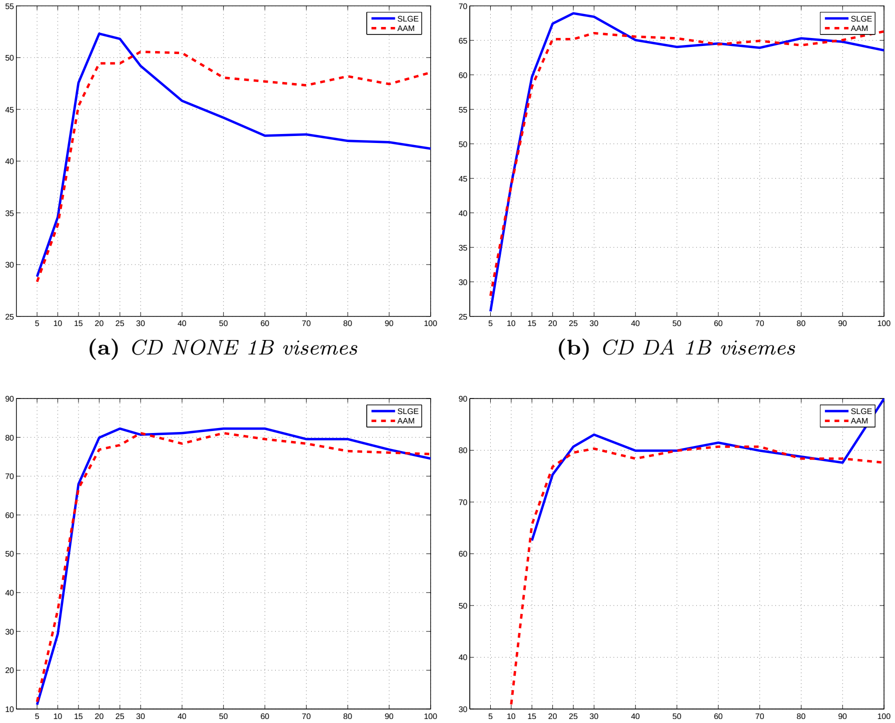
8.3 Influence of High Speed Recordings

In this section we analyse the results as a function of the frame rate of the recorded video. For this we sub-sampled the data to the following levels: 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80 and 90. In the case of the static features the sub-sampling was done at the level of the final feature vectors. However, in the case of the optical features this is not possible and the sub-sampling was performed at the level of the input video stream. The new optical flow fields were computed based on the new video sequences.

In all the experiments analysed we found that below the 10Hz limit the recognition performance is highly degraded and sometimes the systems are not trainable. Above this threshold there is a steep increase in performance that continues until the 25-30Hz level. Above the 30Hz level, as we expected, we found that the high speed recordings have larger influence in the case of high speech rate.

As we mentioned in Section 8.2, the distribution of the speech rate is not uniform over all recognition tasks, and for instance, the digit string recordings were classified entirely as normal speech, while the letter strings recordings had somewhat higher speech rate. It is important to note this because we found out that for normal speech the peak performance is obtained when the recording frame rate is around 30Hz. Moreover, we found that the performance is likely to decrease if a higher recording frame rate is used. Figure 8.1 shows four examples of the performance of the recognisers for four different experiments. Each of the four images in this figure shows a different situation. The image (b) shows a peak at around 25-30Hz then a slight decrease when the frame rate increased. This peak is more visible for the SLGE based features. In the image (c) we see that after 30Hz the performance remains almost constant; sometimes slightly increasing in the interval 50-60Hz. Image (d) shows a strange increase of around 10% at 100Hz. In the image (a) we see a large difference between the performances of the two methods. This difference was found in other experiments as well. It should be noticed that in this image the results are given for the case when only the static features were used. It seems that the SLGE based features are less robust when only the static features are used and the recording frame rate increases. The fact that the performance of the recognisers decreases slightly when the recording frame rate increases is due to the increase of noise in the data. This situation was found more accentuated in the case of the OF based features as shown in Chapter 7. Therefore, we can also conclude that the decrease depends on the robustness of the method used for the parametrization of the input data. With respect to the accuracy of the recognition (Acc), increasing the recording frame rate was found to have a greater effect. We found that the graph of the Acc values has a more prominent reversed J-shape. Figure 8.2 shows some examples.

While in the case of the digit string recognition task we found that a frame rate of 30Hz is optimal, for the letter recognition depending on the proportion of high speech rate the optimal recording frame rate was found to be around the 50Hz level. However, this sometimes depends on the feature extraction method used, namely the results showed that the AAM based systems needed in general a higher frame rate in order to achieve its full potential. The graphs also show that the systems based



(a) *CD NONE 1B visemes*

(b) *CD DA 1B visemes*

(c) *CD DA 1B words*

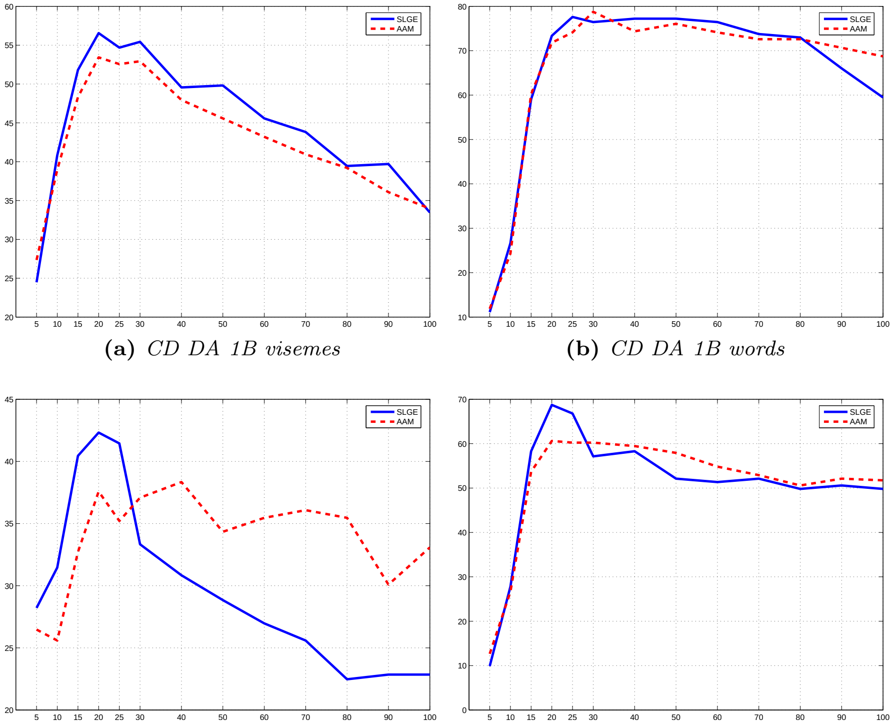
(d) *CD DA 3B words*

Figure 8.1: *Lip reading performance as a function of the recording frame rate. The graphs show the results for both AAM based and SLGE based features. The first line shows the results for digit strings, without or with dynamic features included, for the free grammar case. The second line shows the results for digit strings, at digit level without or with intra-word context, respectively. The x coordinates give the recording frame rate, while the y coordinates give the WRR value of the recognizer.*

on the AAM defined features are more stable when the frame rate increases. Figure 8.3 shows examples for the letter string recognition task. The exception seems to be here the bottom-right image where both systems obtained their maximum at 100Hz level. One more observation is the apparent instability of the SLGE method at higher frame rates, the AAM based approach producing smoother graphs.

The situation is similar in the case of the GU recognition task, as we can see in Figure 8.4.

In order to have the complete picture we include the results for the optical flow defined features as a function of the frame rate. We show first in Figure 8.5 the results for a grammar free recogniser for digits. This example proves that the conclusions from Chapter 7 are valid. The conclusion is that, in the case of the optical flow based features, at high frame rates the insertion error grows rapidly. The WRR figures grow with the frame rate. However, the accuracy of the recognizer decreases



(a) *CD DA 1B visemes*

(b) *CD DA 1B words*

(c) *CD NONE 1B visemes*

(d) *CD NONE 1B words*

Figure 8.2: *Lip reading performance as a function of the recording frame rate. The graphs show the results for both AAM based and SLGE based features. The left-top image shows the results for digit strings, with dynamic features included, for the free grammar case. The right-top image shows the results for digit strings, with dynamic feature included, at digit level without intra-word context. The bottom-left image shows the Acc values for digit strings, without dynamic features included, for the grammar free case, and the bottom-right image shows results for the digit level case. The x coordinate gives the recording frame rate, while the y coordinate gives the Acc value of the recognizer.*

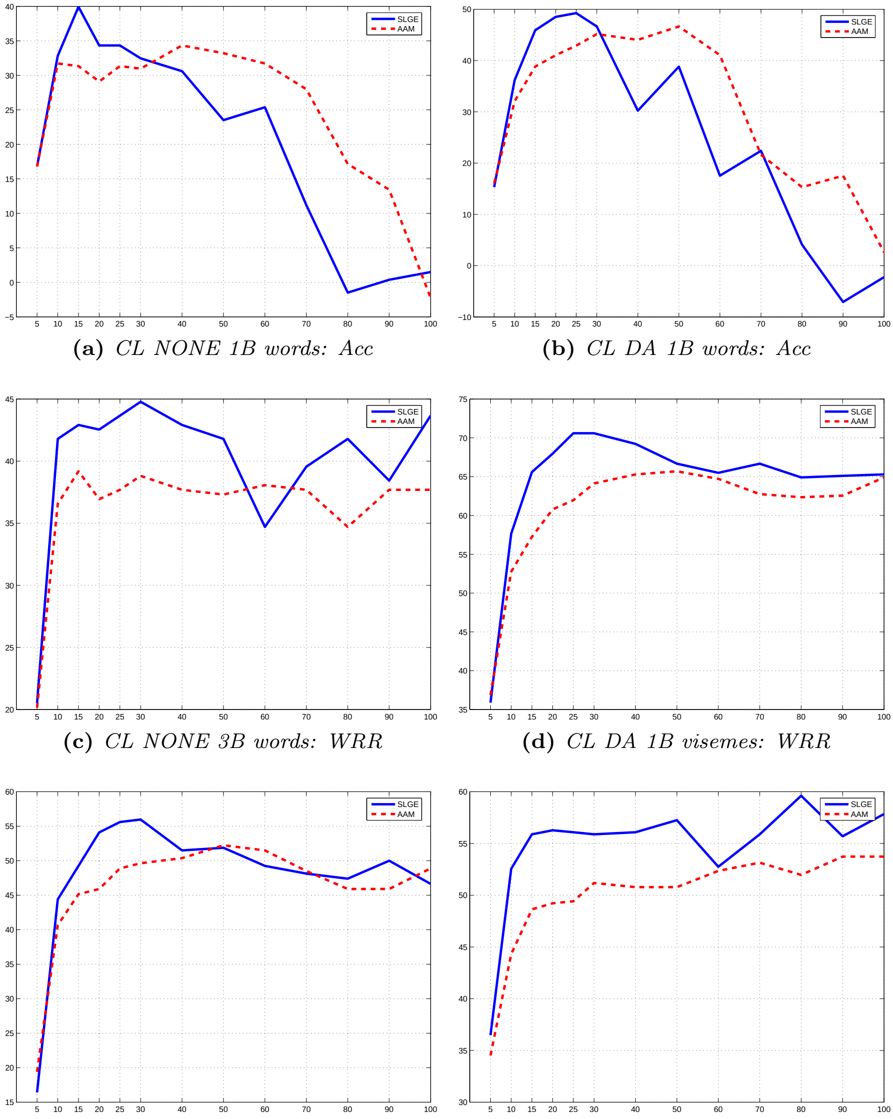
to extremely low levels.

When the system is constrained by using a word network the results get better, therefore, hiding these problems. These results are shown in Figure 8.6. We find in these images that this system peaks at 30Hz level.

8.4 Influence of the Size of the Training Corpus

A basic question on the performance of a system is what is the necessary size of the corpus in order to obtain optimal performance? The answer to this question is not straight forward because it largely depends of the variability of the analysed system.

Based on the model used to approximate the real process we can get some in-



(a) *CL NONE 1B words: Acc*

(b) *CL DA 1B words: Acc*

(c) *CL NONE 3B words: WRR*

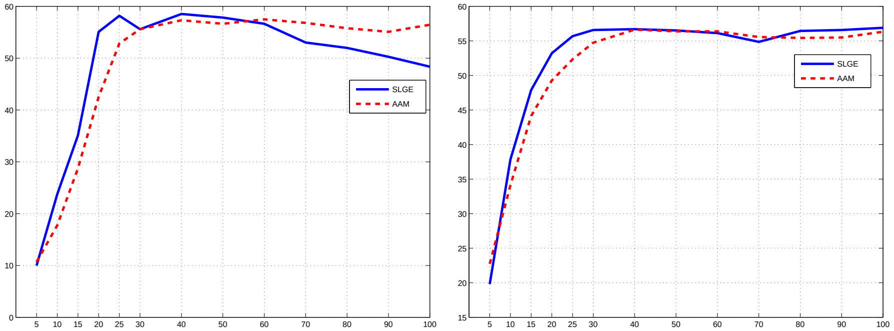
(d) *CL DA 1B visemes: WRR*

(e) *CL DA 1B words: WRR*

(f) *CL NONE 1B visemes: WRR*

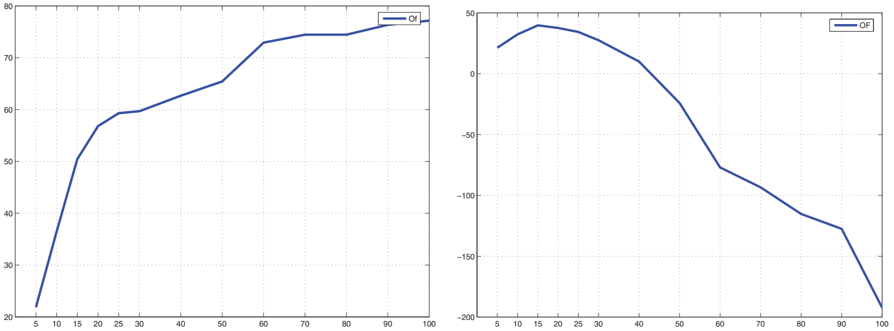
Figure 8.3: *Lip reading performance as a function of the recording frame rate. The graphs show the results for both AAM based and SLGE based features. The images show results for letter strings in different settings. The x coordinate gives the recording frame rate, while the y coordinate gives the WRR or Acc value as is the case of the recognizer.*

sights on this number by counting the number of parameters in the model. It is always necessary to preserve a balance between the model detail and number of the parameters to be estimated. In our case depending on the number of HMM



(a) *GU DA 1B words: WRR* **(b)** *GU DA 1B visemes: WRR*

Figure 8.4: *Lip reading performance as a function of the recording frame rate. The graphs show the results for both AAM based and SLGE based features. The images show results for all grammar generated utterances. The x coordinate gives the recording frame rate, while the y coordinate gives the WRR value.*

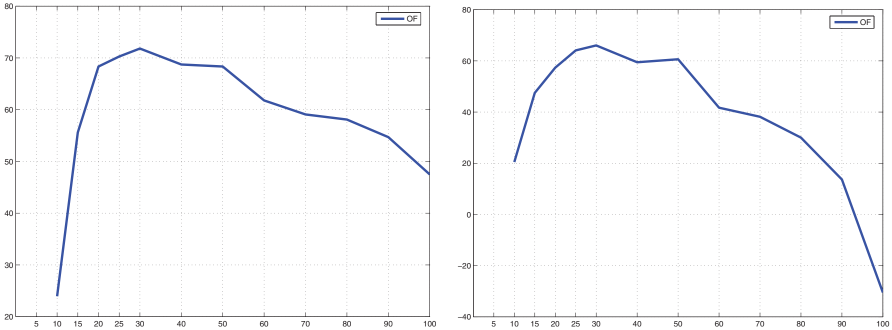


(a) *CD NONE 1B visemes: WRR* **(b)** *CD NONE 1B visemes: ACC*

Figure 8.5: *Lip reading performance as a function of the recording frame rate in the case of OF based features. The images show results for a grammar free connected digit recognizer. The x coordinates give the recording frame rate, while the y coordinates give the WRR or ACC value.*

(e.g. 16 in our case), the structure of the HMM (e.g. three producing states in our case), the number of Gaussian mixtures used, on whether we build context or not and on whether parameter tying was performed and to what degree, the number of parameters to be estimated can count between a few hundred thousand to millions. However, the degree of freedom in the model gives only an indication on the size of the corpus.

The empirical approach is maybe the most reliable one but has very little generalization. This approach consists of building systems on subsets of the same corpus and comparing the results. It is also possible, but unreliable, to compare the results between experiments performed on different corpora. We can compare, for instance,



(a) *CD DA 3B word: WRR* **(b)** *CD DA 3B word: ACC*

Figure 8.6: *Lip reading performance as a function of the recording frame rate in the case of OF based features. The images show results for a grammar free connected digit recognizer. The x coordinates give the recording frame rate, while the y coordinates give the WRR or ACC value.*

the best result reported, as in [Woj03], because the DUTAVSC contains a sub-corpus with respect to the language pool to our corpus. The author reports a maximum performance of 91.1% WRR and 60.0% Acc for a free grammar CD task. These figures compared with our results (i.e 91.39% and 79.43%) suggest a small difference in WRR level but a large one in Acc level. In the constrained grammar task the author reports 86.7% WRR and 81.1% Acc. The comparison is, therefore, misleading because we do not know in what degree the settings used in all these experiments coincide, and also what the influence of the settings is. Therefore, we performed our own experiment using the AAM based approach for the CD and CL tasks. The results on the smaller corpus were published in [Chi09b]. The data corpus used for digits was increased approximately three times, while the corpus for letters was increased two times. The results are summarised in Table 8.1. In both situations we observed a considerable increase in performance. Due to time limitations we did not repeat the experiments with larger corpora to be able to find an upper boundary. This will make the subject of future work.

Table 8.1: *Comparison of the performance with respect to the size of the corpus used (i.e. 90% for training and 10% for testing).*

Number words	WRR	Acc
CD Recognition Task		
883	78.08%	68.49%
2484	91.39%	79.43%
CL Recognition Task		
1564	49.46%	-12.90%
3304	56.34%	17.16%

8.5 Conclusions

We investigated in this chapter the influence of the feature extraction approach used, recognition results as a function of the speaking style and the influence of the high speed recordings on the recognition performances. As we were expecting, the speaking style has a large influence on the recognition systems. We found that for higher speech rates we need to use higher sampling rates. However, as was shown in this chapter the 100Hz level is usually too high and that 60Hz covers all possible situations. What was however surprising, was that having a higher frame rate not only involves larger processing times with no visible increase in performance, but when used for inappropriate situation and with inappropriate parametrization it can largely decrease the performance of the system sometimes to the point where the systems do not train anymore. This is due to the fact that at higher frame rates the probability of introducing noise in the system is larger. As we could see in this chapter, this is also correlated with the feature extraction method used.

Another result of the analysis shown in Section 8.3 is that the performance figures shown in the previous chapters were all the time underestimated, since the results shown there were based on 100Hz recordings, which are much lower than the peak results. This is especially true for the accuracy figures.

The most important conclusion is that high speed recordings are necessary but only for the case when we deal with high speech rate. However, in normal situations the amount of high speech rate cannot be a-priory known and a distribution over the speech rate might at best be computed. The problem is that using high speed recordings for normal speech might be detrimental to the performance and, therefore, its use should be carefully evaluated. Nevertheless, in the situations where the incidence of high speech rate is large, it brings a plus of performance. These situations are not necessarily improbable, we can for instance think of sports live commentators and auctioneers who most of the time use a very high speech rate.

Chapter 9

Conclusions: Summing Up, General Thoughts and Future Directions

The main goal of this thesis was to investigate the limitations of the current approaches and the state of the art, and moreover to investigate the possibilities to boost lip reading performance toward a robust system. We approached our goal by splitting it into more specific research questions that combined give the answer to the main research question.

Are we able to build a lip reader for the Dutch language? Is the Hidden Markov Model suitable for this task?

We presented in this thesis the performance of various lip readers for the Dutch language built based on three different data parametrization methods. We obtained good performance for recognition tasks such as connected digits (WRR 91.39%), and connected letters (WRR 56.34%), similar to the performances obtained by other researchers for English or other languages. What is more rewarding is that we obtained good results for more complex recognition tasks such as grammar based utterances (WRR 56.89%) and random sentences (WRR 39.23%). The systems built for these tasks obtained greater performance than previous work.

With respect to the inference method used, we knew from the beginning that the Hidden Markov Models approach is the most successful approach for speech recognition. As seen from our successes as presented in this thesis we can conclude that the HMM approach gives similar performances in the case of lip reading. However, more time should be invested in choosing the structure of the silence models.

What are the problems with the existing data corpora for lip reading? Can we write a set of guidelines for building a data corpus for lip reading? Can we create a corpus for Dutch language large enough to support our endeavour?

These questions found their answer in Chapters 2 and 4. We started our research with a complete investigation of the existing data corpora. We compiled a set of rules and directions for building a data corpus and built a very large data corpus for lip reading in the Dutch language.

A short summary of the guiding rules can be found below:

1. Carefully state the recognition tasks and make sure that all are equally represented in the resulting corpus.

2. Choose the respondents carefully and record their data for later use.

3. Choose the language data carefully and state the tasks considered.

4. Consider different speaking styles.

5. The Region Of Interest (ROI) should be as close to the speaker's face as possible. Make sure the images show good detail around the speaker's mouth.

6. Use a sufficiently high video recording speed. Our analyses showed that 60Hz is a good number.

7. Decide upon the recording environment and make sure it is consistent during recordings.

8. Automate as much as possible the data acquisition process.

9. Make sure the post processing of the corpus is performed in a consistent manner.

The value of this corpus was shown extensively during our experiments. The new corpus strengths were the high speed recordings (i.e. 100Hz), the dual synchronized view (i.e. we recorded side and frontal view; we recorded dual view because we wanted the resulting corpus be used in future research experiments), the large spectrum of recognition tasks targeted, the stated recognition of different speech styles and the great coverage of the language and speaker variations. We do not see this corpus as finished though, we actually see it as a first step towards an ever growing source of valuable data for lip reading. We think that it is worth mentioning one more time that a good data corpus is of paramount importance in a data hungry problem as lip reading.

What is the influence of the data parametrization method on the performance of lip reader? What are the strengths of the various feature extraction methods?

Each parametrization method was presented in a separate chapter and their strengths and weaknesses were discussed in the corresponding chapter and summarised in the local conclusion sections. The parametrization methods were chosen to cover all three types of feature extraction techniques: namely geometric, appearance based and combined. Other aspects were also taken into account. For instance we used a model based approach (AAM) and a model free approach (SLGE). We also investigated the optical flow analysis for lip reading which is a method that tries to recover the actual movement on the speaker's face as opposed to approximating the motion by computing the first and second derivative.

A summary of the strengths and weaknesses is given below:

1. The optical flow directly describes the motion on the speaker's face.
2. The optical flow analysis does not produce the same performance as the other methods due to the poor performance of the optical flow detection algorithm.
3. The optical flow was the most computational demanding from the three approaches.
4. The other two approaches performed somewhat similar. However, the AAM approach performed better at higher frame rates.
5. Both SLGE and AAM methods were able to perform in real time.
6. The SLGE approach included some appearance based features which increased its performance.

Where possible, we investigated a number of improvements that made their application more robust such as outlier removal and ROI detection. The performance of the methods was investigated both from the point of view of the computer vision (CV) abilities but also from the point of view of the final system recognition performances. We gradually showed the possible steps towards a more robust lip reader. The main issue with the data parametrization is the poor generalization of the CV algorithms. In most cases the applicability of the detection and tracking methods is reduced to laboratory, controlled conditions. The algorithms are adapted towards the conditions of the given data corpus and have a small generalization to different environments. Even with the introduction of newer and more powerful algorithms, the state of the art in CV in the visual domain is still a long way from the capacity of humans. For instance the two great problems for CV, the variance in illumination and occlusions, are still plaguing the process of detection and tracking. In this domain there is a great need of a revolutionary technique which can bring the artificial systems closer to the human capacities. One direction which we think is necessary is the understanding of the human visual perception process and building an automated replica of this process. A colour space which conserves the perception uniformity corroborated with an adequate model for human perception is in our opinion a necessity towards Robust Computer Vision. An approach based on, or similar with, the Gestalt psychology theory [Ste03] of the brain and human visual perception seems in our opinion a very strong starting point, even though the computing power available might be still not sufficient for this type of approach. In Chapter 8 we compared the three methods from the point of view of the induced performance of the resulting lip readers. We can clearly see there that, even though all three methods have good results comparable if not sometimes greater than the previous results, there are still great differences between these systems. Their performances differ under different conditions with no one appearing as a clear winner. This observation, justifies the current debate for the best data parametrization.

Should we concentrate on the contour of the mouth, or is it better to have a broader appearance based approach?

The answer to this question is not 100% clear from the results. The pure appearance based, optical flow approach, performed less impressive than the other two approaches. However, the accuracy of the optical flow features depends heavily on the performance of the optical flow detection algorithm. Also, the SLGE approach also incorporated some appearance based features and the geometric features computed based on the AAM results contained a more complete description of the lips. What we can say is that the geometric AAM based features were more stable at higher frame rates but that it also needed higher frame rates to achieve its maximum potential. Our conclusion is that we should use a combination of the two approaches. However, we should carefully choose the appearance based features because they are prone to bring a large amount of noise in the data.

How important is the motion? Do we need to compute the motion flow on the speaker face or is sufficient to generate a static model of the speaker face and only compute the derivatives based on this model?

With some disappointment we concluded that the optical flow approach did not perform according to our expectations. However, some of the loss in performance can be attributed to the inaccuracies of the optical flow detection algorithm and not to the limited information content of the features. Since it is very difficult to compute the ground truth of the optical flow on the speaker's face, this argument is not easily verified. On the other side we found that the addition of the first and second derivative to the static features greatly improves the performance, but not in the case of the optical flow features where it does not bring much improvement.

What is the influence of the speaking style on the performance of the lip reader?

We investigated the influence of the speaking rate on the lip reading performance. In order to do this we included in the data corpus utterances for which the speakers were instructed to alter their speaking style. We found that not all speakers were able to increase their speaking rate, but also that the normal speech rate for some speakers' was faster than the high speech rate of other. We used the number of words per minute as a measure of the speaking rate. We concluded that the speaking style has a great impact on the recognition performance, but that the problem should be carefully approached. As expected for faster speech rate we needed to use a higher recording rate in order to obtain the maximum of performance. In order to solve this problem, the system can actively adapt to the speaking style of the user by for instance increasing the video recording frame rate. Also, the system can adapt the language model to reflect the characteristics of the high speech rate. For this, however, it is necessary to understand the transformation the language suffers when speaking fast (e.g. shorter vowels, concatenated words, etc.).

Is there any correlation between the speaking style and the choice of the data parametrization used, namely is one parametrization more suitable for some speaking environ-

ment than the others?

The conclusion we have drawn in this respect is that for high speech rate we need a higher recording rate in order to achieve the maximum performance. We also observed that the geometric features computed based on the AAM approach were somewhat more stable at higher frame rates. From this we can conclude that these features should be preferred to the SLGE features.

Do we need to use a higher recording rate on the visual side?

We concluded that it is not necessary to use 100Hz frame rate. The performance of the lip reader actually decreases due to the increased noise at this high rate. This was an unexpected finding of our analysis. The use of high speed recordings should be done selectively, because the different speaking styles have different peaks and an inappropriate usage can do more damage than good. For normal speech rate we found that the optimum recording sampling rate is around the 30Hz value. For high speech rate, however, the recognition performance peaks in the interval 40-60Hz, but this is sometimes dependent on the data parametrization technique.

What is the influence of the size of the corpus on the performance of the resulting recognition system?

It is difficult to answer this question by comparing the results on different corpora. Therefore, in order to get some quantitative insight on the influence of the corpus size on the performance we compared the results of several systems trained on the complete corpus with the results of the same systems trained on a subset of the corpus. Being a Bayesian framework we expected that having more data would bring more information about the distributions that govern the modelled process. This was the case here. We found improved performance in all tests we performed when the larger corpus was used. However, it is still difficult to estimate the amount of data necessary to achieve the maximum, since the results do not show a clear pattern that unifies all the results. In our opinion an exact answer is only found by trial and error. However, the conclusion: "more data, better results" still stands.

What is the best definition of the visemes?

To answer this question we investigated scientific papers that covered this topic for the Dutch language. We think that lip reading would gain substantially if an independent definition for visemes would be found. On top of the fact that the visual speech is poorer than the aural speech comes also the poor definition of the visemes. This further reduces the performance of automatic lip reading.

For future work we see two immediate improvements to the current work, namely improved language models and improved silence models.

As we have seen in the analyses of the results in some experiments, we recorded high insertion errors. This error can be explained by poorly trained silence models and, therefore, we consider that the silence models should be further developed after a careful analysis of the exact influence on the performance. Another solution would be to use a separate model for detecting the onset and offset in addition to better trained silence models.

With respect to the language model we noticed that the language models we used based on bi-grams were not sufficiently strong for the case of continuous speech because they were trained on a very small corpus. We already started to gather more language data for training more advanced tri-grams and four-grams language models but at this time building new language models was not within the scope of this thesis.

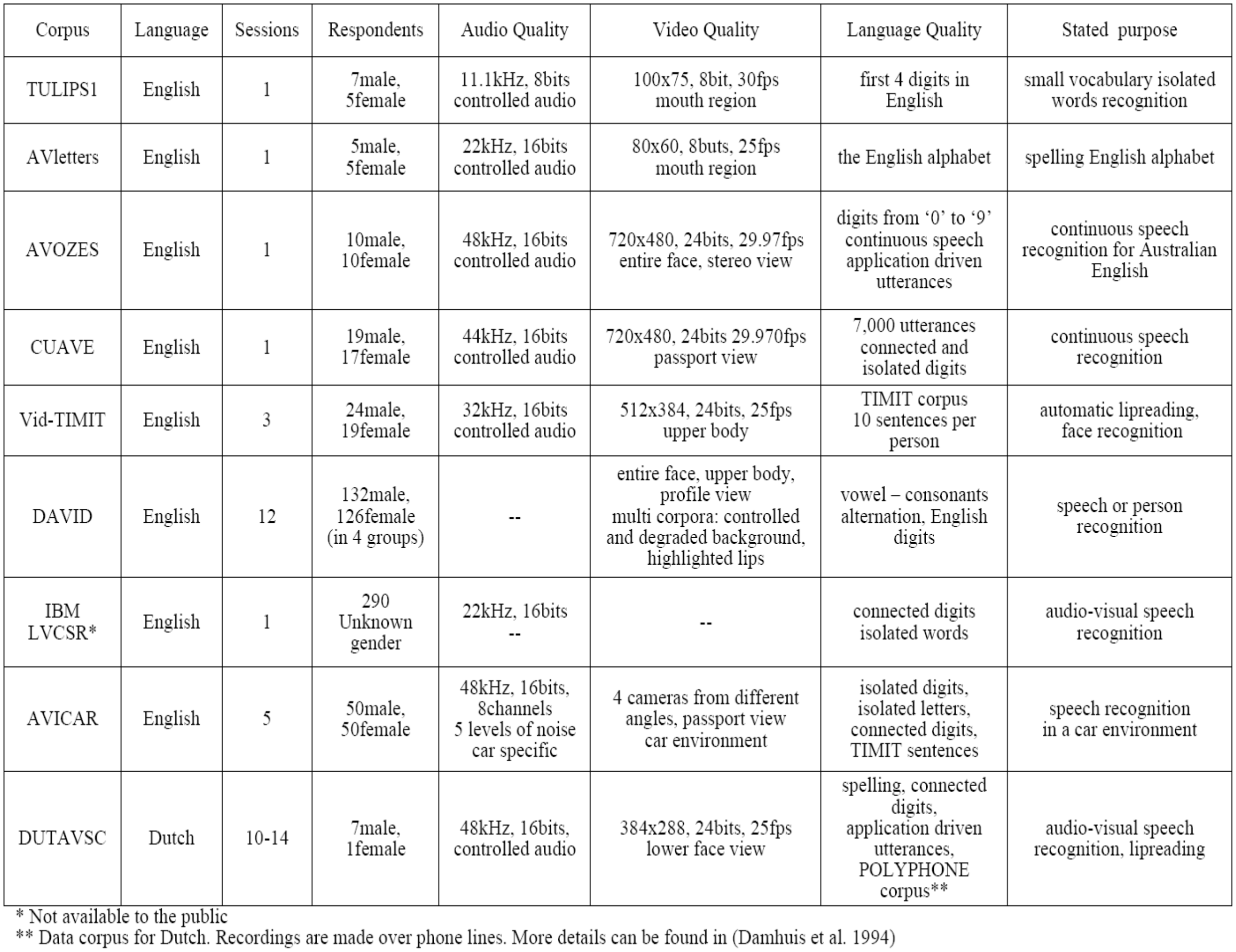
Other future developments we envision are side view lip reading and 3D model lip reading, both supported by our corpus. Neither of these two approaches constitute new ideas (e.g. see [Yos04; Chi08b] for silhouette lip reading and [Cıs04] for 3D lip reading). However, we consider that a careful analysis based on our corpus should be done.

To conclude, at this moment lip reading is making progress both in accuracy and robustness, but unlike speech recognition it still has some big thresholds to surmount. However, these problems are common to a large number of applications, some of them like emotion recognition or computer vision in general having currently a larger momentum. This coupling will assure that in the near future lip reading could make use of newer, more advanced technologies. It is our belief that lip reading will eventually become a major addition to future user interfaces which will bring additional capabilities but also more importantly a large relief to the human users. Eventually, a holistic approach which will include all other connected domains will make the user interfaces of the future.

Appendix A

Data Corpora for Lip Reading

This appendix contains the principal characteristics of the existing data corpora for lip reading at the time when the current dissertation was written.



Corpus	Language	Sessions	Respondents	Audio Quality	Video Quality	Language Quality	Stated purpose
TULIPS1	English	1	7male, 5female	11.1kHz, 8bits controlled audio	100x75, 8bit, 30fps mouth region	first 4 digits in English	small vocabulary isolated words recognition
AVletters	English	1	5male, 5female	22kHz, 16bits controlled audio	80x60, 8buts, 25fps mouth region	the English alphabet	spelling English alphabet
AVOZES	English	1	10male, 10female	48kHz, 16bits controlled audio	720x480, 24bits, 29.97fps entire face, stereo view	digits from '0' to '9' continuous speech application driven utterances	continuous speech recognition for Australian English
CUAVE	English	1	19male, 17female	44kHz, 16bits controlled audio	720x480, 24bits 29.970fps passport view	7,000 utterances connected and isolated digits	continuous speech recognition
Vid-TIMIT	English	3	24male, 19female	32kHz, 16bits controlled audio	512x384, 24bits, 25fps upper body	TIMIT corpus 10 sentences per person	automatic lipreading, face recognition
DAVID	English	12	132male, 126female (in 4 groups)	--	entire face, upper body, profile view multi corpora: controlled and degraded background, highlighted lips	vowel – consonants alternation, English digits	speech or person recognition
IBM LVCSR*	English	1	290 Unknown gender	22kHz, 16bits --	--	connected digits isolated words	audio-visual speech recognition
AVICAR	English	5	50male, 50female	48kHz, 16bits, 8channels 5 levels of noise car specific	4 cameras from different angles, passport view car environment	isolated digits, isolated letters, connected digits, TIMIT sentences	speech recognition in a car environment
DUTAVSC	Dutch	10-14	7male, 1female	48kHz, 16bits, controlled audio	384x288, 24bits, 25fps lower face view	spelling, connected digits, application driven utterances, POLYPHONE corpus**	audio-visual speech recognition, lipreading

* Not available to the public

** Data corpus for Dutch. Recordings are made over phone lines. More details can be found in (Damhuis et al. 1994)

Figure A.1: *Comparison of the major existing data corpora which can be used for lip reading.*

Appendix B

Grammars for Recognition Tasks

This appendix contains the listings of the grammar rules used for defining the digit strings, the letter strings and the complete grammar set recognition tasks used during the experiments.

B.1 Digit String Recognition Task(CD)

```
$digit = \<0\> | \<1\> | \<2\> | \<3\> | \<4\> | \<5\> | \<6\> |
         \<7\> | \<8\> | \<9\>;
$eightdigits = $digit  $digit  $digit  $digit
               $digit  $digit  $digit  $digit;
$connectedDigits = ( $digit | $eightdigits );
(     SENT-START $connectedLetters SENT-END     )
```

B.2 Letter String Recognition Task(CL)

```
$letter = \<A\> | \<B\> | \<C\> | \<D\> | \<E\> | \<F\> | \<G\> |
          \<H\> | \<I\> | \<IJ\> | \<J\> | \<K\> | \<L\> | \<M\> |
          \<N\> | \<O\> | \<P\> | \<Q\> | \<R\> | \<S\> | \<T\> |
          \<U\> | \<V\> | \<W\> | \<X\> | \<Z\>;
$eightletters = $letter  $letter  $letter  $letter
                $letter  $letter  $letter  $letter;
$connectedLetters = ( $letter | $eightletters );

(     SENT-START $connectedLetters SENT-END     )
```

B.3 Complete Grammar Set Recognition Task(GU)

```
$digit = \<0\> | \<1\> | \<2\> | \<3\> | \<4\> | \<5\> | \<6\> |
         \<7\> | \<8\> | \<9\>;
$eightdigits = $digit  $digit  $digit  $digit
               $digit  $digit  $digit  $digit;
```

```
$connectedDigits = $digit | $eightdigits;
$zeventig = zeventig | zeuventig;
$numberdigits_01 = vier;
$numbertens = twintig | dertig | veertig | vijftig | zestig |
              $zeventig | tachtig | negentig;
$number20_99 = achtendertig | achtenveertig | vierenvijftig |
               zeuvenenzestig;
$number100s = driehonderd | zeuvenhonderd | achthonderd |
              negenhonderd;
$number =
    $numberdigits_01 |
    $number20_99 |
    $number100s |
    $number100s ( $number20_99 | $numberdigits_01);

$amount = $number (euro | euro’s);
$greeting = goedemorgen | goedemiddag | goedenavond;
$please = alstublieft | alsjeblieft;
$want = wil | wilde ;
$accounttype = (rekening | privebankrekening);
$accountnumber = [nummer] $digit $digit $digit $digit
                          $digit $digit $digit $digit;
$my = [mijn];
$account = $accounttype [$accountnumber];

$action = $amount van [$my] $account [naar [$my] $account]
          overmaken |
    $amount op [$my] $account storten |
    $amount storten op [$my] $account |
    $amount opnemen van [$my] $account |
    $amount van [$my] $account opnemen;

$bankingSentences = ([$greeting] ik $want [graag]
                      $action [$please] |
    [$greeting] ik ($want | zou) $action graag |
    [$greeting] ik zou graag $action [$please]);

$word = Antarctica | Britse | Poolse | Tilburg | aard |
        achtergronden | afdruk | afsluiten | avontuur |
        beurs | cheques | commentaar | deelnamenummer |
        donderdag | gebeurd | geheim | gooitje | jurk |
        kroon | langer | leider | lekker | mens |
        ontboden | ontwikkeld | opstanding | overgebracht |
        paar | registratie | rijk | schuin | stampvol |
        standaard | stem | tevergeefs | vanuit | verband |
        veroorzaakten | vier | vijfdaagse | vliegtuigongeluk |
        voorbarige | wenden | wetenschapswinkel | woordvoerder |
        zeventigjarige | zure;
$eightwords = $word  $word  $word  $word
              $word  $word  $word  $word;
$connectedWords = ( $eightwords);

$letter = \<A\> | \<B\> | \<C\> | \<D\> | \<E\> | \<F\> | \<G\> |
          \<H\> | \<I\> | \<IJ\> | \<J\> | \<K\> | \<L\> | \<M\> |
          \<N\> | \<O\> | \<P\> | \<Q\> | \<R\> | \<S\> | \<T\> |
          \<U\> | \<V\> | \<W\> | \<X\> | \<Z\>;
$eightletters = $letter $letter $letter $letter
                $letter $letter $letter $letter;
$connectedLetters = ( $letter | $eightletters );

(     SENT-START ( $connectedLetters |
                  $connectedDigits |
                  $bankingSentences |
                  $connectedWords)
      SENT-END      )
```

Appendix C

Utterance Types

This appendix contains the lists with the utterance types and their descriptions as provided to the respondents during the recording sessions for the NDUTAVSC corpus.

Table C.1: *Utterance Types for the First Recording Sessions: 64 utterances.*

Times	Speaking style	Description
3	Normal speech rate	Random digit sequences of length 8
3	Fast speech rate	—//—
3	Whispering	—//—
3	Normal speech rate	Spelling a random word of variable length
3	Whispering	—//—
3	Normal speech rate	Lists of random words of length 8
3	Fast speech rate	—//—
3	Whispering	—//—
5	Normal speech rate	Fixed grammar bank application sentences
5	Fast speech rate	—//—
5	Whispering	—//—
5	Normal speech rate	Random sentences taken from Polyphone
5	Fast speech rate	—//—
5	Whispering	—//—
5	Normal speech rate	Every day used expressions
5	Normal speech rate	Short answers to random open questions

Table C.2: *Utterance Types for the Second Recording Sessions: 125 utterances.*

Times	Speaking style	Description
45	Normal speech rate	Sentences taken from Polyphone
10	—//—	Random digit strings of length 8
10	—//—	Random letter strings of length 8
30	—//—	Isolated digits
30	—//—	Isolated letters

Table C.3: *Utterance Types for the Third Recording Sessions: 155 utterances.*

Times	Speaking style	Description
45	Normal speech rate	Sentences taken from Polyphone
10	—//—	Random digit strings of length 8
30	—//—	Random letter strings of length 8
20	—//—	Isolated digits
50	—//—	Isolated letters

Table C.4: *Utterance Types for the Fourth Recording Sessions: 160 utterances.*

Times	Speaking style	Description
60	Normal speech rate	Sentences taken from Polyphone
40	—//—	Random digit strings of length 8
40	—//—	Random letter strings of length 8
10	—//—	Isolated digits
10	—//—	Isolated letters

Appendix D

Utterances Sample

This appendix gives an example of the utterances file as presented to the speaker during a recording session for the NDUTAVSC corpus by the prompter tool.

```
Utt. type: 15 Description: Beantwoord de volgende vragen
                           zo kort mogelijk.
  1     :-> Wat is uw studentnummer (indien van toepassing)?
  3     :-> Kunt u uw naam spellen?
  4     :-> Wat is uw studentnummer (indien van toepassing)?
  5     :-> Hoe lang verwacht u nog te moeten studeren (indien van toepassing)?
Utt. type: 2 Description: Fluister de volgende cijfercombinaties
                           op normaal tempo.
  6     :-> <1> <0> <3> <1> <6> <5> <0> <5>
  7     :-> <8> <8> <1> <3> <4> <4> <0> <9>
  8     :-> <9> <2> <3> <5> <8> <7> <7> <4>
Utt. type: 1 Description: Lees de volgende cijfercombinaties
                           op, maar sneller dan gewoonlijk.
  9     :-> <0> <9> <3> <5> <8> <2> <0> <5>
  10    :-> <5> <9> <1> <2> <7> <4> <7> <7>
  11    :-> <0> <9> <5> <7> <1> <5> <1> <1>
Utt. type: 4 Description: Fluister en spel de volgende woorden
                           op normaal tempo.
  12    :-> <g> <e> <m> <e> <e> <n> <t> <e>
  13    :-> <d> <r> <u> <k> <k> <e> <n>
  14    :-> <g> <e> <l> <i> <j> <k>
Utt. type: 3 Description: Spel de volgende woorden op normaal tempo.
  15    :-> <s> <c> <h> <r> <i> <j> <v> <e> <r>
  16    :-> <r> <u> <i> <t> <e> <r>
  17    :-> <z> <i> <e> <k> <e> <n> <h> <u> <i>
            <s> <s> <p> <u> <l> <l> <e> <n>
Utt. type: 6 Description: Lees de volgende woordenlijsten voor,
                            maar sneller dan gewoonlijk.
  18    :-> <ontstaansgeschiedenis> <kopen> <vrouw> <commentaar> <Ypsilon>
            <hield> <vakantiereisje> <gelegen>
  19    :-> <Utrecht> <ongelukje> <gebruik> <wetenschapswinkel> <brengen>
            <kloppen> <gezet> <voorstel>
  20    :-> <vanuit> <zure> <tevergeefs> <veroorzaakten> <commentaar>
            <gebeurd> <geheim> <avontuur>
Utt. type: 7 Description: Fluister de volgende woordenlijsten
                           op normaal tempo.
  21    :-> <zeven> <programmaboekje> <zorgvuldige> <zevende>
             <februari> <beter> <gestationeerde> <cent>
  22    :-> <lekker> <ontboden> <voorbarige> <stem> <kroon> <gooitje>
             <jurk> <wetenschapswinkel>
  23    :-> <standaard> <Antarctica> <Poolse> <zeventigjarige> <beurs>
          <donderdag> <overgebracht> <vijfdaagse>
Utt. type: 5 Description: Lees de volgende woordenlijsten
                             voor op normaal tempo.
  24    :-> <afsluiten> <cheques> <verband> <opstanding> <wenden>
          <deelnamenummer> <achtergronden> <rijk>
  25    :-> <wenden> <registratie> <leider> <aard> <vliegtuigongeluk>
          <vier> <ontwikkeld> <afdruk>
  26    :-> <mens> <schuin> <paar> <woordvoerder> <langer> <Tilburg>
          <stampvol> <Britse>
Utt. type: 8 Description: Spreek de volgende zinnen uit op normaal tempo.
  27    :-> Goedenavond ik wil 770 euro's van mijn privebankrekening
          opnemen graag.
  28    :-> Goedenavond ik wilde 760 euro van mijn bankrekening <6> <1>
          <7> <8> <0> <8> <0> <4> overmaken alsjeblieft.
  29    :-> Goedenavond ik wilde 760 euro van mijn bankrekening <6> <1>
          <7> <8> <0> <8> <0> <4> overmaken alsjeblieft.
  30    :-> Ik wilde 354 euro op mijn priverekening <3> <9> <4> <7> <2>
          <6> <4> <7> storten.
  31    :-> Goedemorgen ik wilde 348 euro storten op mijn bankrekening
          alstublieft.
Utt. type: 10 Description: Fluister de volgende zinnen op normaal tempo.
  32    :-> Goedenavond ik wilde 890 euro van bankrekening nummer <9>
          <2> <8> <0> <5> <4> <2> <4> naar mijn priverekening <1>
          <8> <2> <1> <0> <9> <1> <0> overmaken.
  33    :-> Ik wilde 38 euro storten op bankrekening <6> <7> <1> <7>
          <5> <0> <3> <2> alsjeblieft.
  34    :-> Goedemiddag ik wilde 67 euro op mijn privebank rekening <8>
          <6> <3> <4> <5> <7> <1> <5> storten.
  35    :-> Ik wilde 50 euro op mijn rekening <9> <5> <3> <0> <2> <4>
          <9> <5> storten alsjeblieft.
  36    :-> Goedenavond ik wil 50 euro storten op mijn bankrekening
          graag.
Utt. type: 9 Description: Spreek de volgende zinnen uit,
                             maar sneller dan gewoonlijk.
  37    :-> Goedenavond ik wil 970 euro van bankrekening <6> <5> <7>
          <8> <0> <4> <2> <1> naar priverekening <4> <5> <6> <5> <4>
          <9> <1> <9> overmaken graag.
  38    :-> Goedenavond ik wilde 980 euro's van bankrekening <7> <9>
          <8> <1> <6> <4> <8> <7> opnemen alsjeblieft.
  39    :-> Ik zou graag 20 euro van mijn rekening overmaken.
  40    :-> Goedemiddag ik wilde 60 euro van mijn bankrekening <2> <4>
          <3> <0> <5> <7> <6> <2> naar mijn bankrekening <9> <5> <9>
          <3> <9> <2> <4> <6> overmaken alstublieft.
  41    :-> Ik zou 150 euro van mijn privebankrekening <6> <3> <8> <7>
          <7> <7> <7> <9> opnemen graag.
Utt. type: 12 Description: Spreek de volgende zinnen uit,
                             maar sneller dan gewoonlijk.
  42    :-> Toch dient zich nu een andere oplossing voor duikvereniging
          aan.
  43    :-> Alle spelers dragen hetzelfde blauwe uniform.
  44    :-> Het fleurige boeket bloemen hing er na drie dagen al
          helemaal verlept bij.
  45    :-> De verzekeringsmaatschappij keerde een half miljoen euro
          uit.
  46    :-> De burgemeester van Cuijk legt op een maart 1988 zijn
          functie neer.
Utt. type: 11 Description: Spreek de volgende zinnen uit op normaal tempo.
  47    :-> Bij de sluiting van de drugspanden worden veel aanhoudingen
          verricht.
  48    :-> Je zou kunnen zeggen dat ze meer een met de rolstoel worden.
  49    :-> Naar verluidt komen zij over drie maanden terug.
  50    :-> Bij de vereniging zijn kaarten in omloop met een waarde van
          6.000 euro.
  51    :-> Het maken van winst was volgens de bestuurder niet de
          eerste prioriteit.
Utt. type: 13 Description: Fluister de volgende zinnen op normaal tempo.
  52    :-> Jouw broer heeft mij een brief gestuurd om mij te bedanken
          voor de chocolaatjes.
  53    :-> Meer dan duizend beroepsmilitairen werden in een enquete om
             hun mening gevraagd.
  54    :-> De ruiter kon zijn paard voor de hoge hindernis niet meer
             in bedwang houden.
  55    :-> Na tien jaar is de grootse operationele melkreclamecampagne
             gestopt.
  56    :-> Ik heb er moeite mee om zijn excuses zonder meer te
             aanvaarden.
Utt. type: 14 Description: Spreek de volgende alledaagse
                    uitdrukkingen uit op normaal tempo.
  57    :-> Goedemorgen!
  58    :-> Pardon.
  59    :-> Goeiedag!
  60    :-> Dat had ik niet zo bedoeld.
  61    :-> Dankuwel.
Utt. type: 0 Description: Lees de volgende cijfercombinaties
                            voor op normaal tempo.
  62    :-> <9> <1> <3> <8> <4> <2> <8> <3>
  63    :-> <3> <9> <1> <3> <7> <8> <3> <8>
  64    :-> <4> <7> <7> <5> <5> <8> <4> <6>
```

Bibliography

[Ana89] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, vol. 2:pp. 283–310, 1989.

[Ars06] I. Arsic and J.-P. Thiran. Mutual information eigenlips for audiovisual speech recognition. In *14th European Signal Processing Conference (EUSIPCO)*. 2006.

[Att06] N. van Atteveldt. *Speech meets script fMRI studies on the integration of letters and speech sounds*. Ph.D. thesis, Universiteit Maastricht, 2006.

[Bak75] J. Baker. The DRAGON system–An overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23(1):pp. 24–29, 1975.

[Bak07] S. Baker, D. Scharstein, and J. Lewis. A database and evaluation methodology for optical flow. In *Computer Vision (ICCV 2007), Eleventh IEEE International Conference on*. October 2007.

[Bar94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, vol. 12(1):pp. 43–77, February 1994.

[Bau66] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, vol. 37:pp. 1554–1563, 1966.

[Bau67] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull Amer Math Soc*, vol. 73(3):pp. 360–363, 1967.

[Bau68] L. E. Baum and G. R. Sell. Growth functions for transformations on manifolds. *Pac J Math*, vol. 27(2):pp. 211–227, 1968.

[Bau70] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, vol. 41(1):pp. 164–171, 1970.

[Bau72] L. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, vol. 3(1):pp. 1–8, 1972.

[Beu96] D. Beun. *Viseme syllable sets.* Master's thesis, Institute of Phonetic Sciences, University of Amsterdam, 1996.

[Bla99] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1999.

[Boo94] T. Boogaart, L. Bos, and L. Bouer. Use of the dutch polyphone corpus for application development. In *2nd IEEE Workshop on Iterative Voice Technology for Telecomunication Applications.* September 1994.

[Boo96] F. Bookstein. Landmark methods for forms without landmarks: localizing group differences in outline shape. *San Francisco, CA, USA*, pp. 279–289, 1996.

[Bre85] M. Breeuwer. *Speechreading Suplimented With Auditory Information.* Ph.D. thesis, Free University of Amsterdam, 1985.

[Bre93] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 1. Institute of Electrical Engineers Inc (IEE), 1993.

[Bre94] C. Bregler and Y. Konig. "Eigenlips" for robust speech recognition. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94 IEEE International Conference on.* 1994.

[Bro04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Lecture notes in computer science*, pp. 25–36, 2004.

[Bru05] A. Bruhn and J. Weickert. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *International Journal of Computer Vision*, vol. 61(3):pp. 211–231, 2005.

[Buc07] J. N. Buchan, M. Pare, , and K. G. Munhall. Spatial statistics of gaze fixations during dynamic face processing. *Social Neuroscience*, vol. 2(1):pp. 1–13, 2007.

[Chi96] C. Chibelushi, S. Gandon, J. Mason, F. Deravi, and R. Johnston. Design issues for a digital audio-visual integrated database. In *Integrated Audio-Visual Processing for Recognition, Synthesis and Communication (Digest No: 1996/213), IEE Colloquium on*, pp. 7/1–7/7. 1996.

[Chi97] G. I. Chiou and J. N. Hwang. Lipreading from color video. *IEEE Transactions on Image Processing*, vol. 6(8):pp. 1192–1195, 1997.

[Chi07a] A. G. Chiţu and L. J. M. Rothkrantz. Building a Data Corpus for Audio-Visual Speech Recognition. In *Proceedings of Euromedia2007*, pp. 88–92. 2007.

[Chi07b] A. G. Chiţu and L. J. M. Rothkrantz. The Influence of Video Sampling Rate on Lipreading Performance. In *12-th International Conference on Speech and Computer (SPECOM'2007)*, pp. 678–684. Moscow, October 2007.

[Chi07c] A. G. Chiţu, L. J. M. Rothkrantz, P. Wiggers, and J. C. Wojdel. Comparison between different feature extraction techniques for audio-visual speech recognition. *Journal on Multimodal User Interfaces*, vol. 1(1):pp. 7–20, March 2007.

[Chi08a] A. G. Chiţu and L. J. M. Rothkrantz. Dutch Multimodal Corpus for Speech Recognition. In *LREC 2008 Workshop on Multimodal Corpora*, pp. 56–59. ELRA, May 2008.

[Chi08b] A. G. Chiţu and L. J. M. Rothkrantz. On Dual View Lipreading Using High Speed Camera. In *Euromedia'2008*, pp. 43–51. Eurosis, 2008.

[Chi09a] A. G. Chiţu and L. J. M. Rothkrantz. The New Delft University of Technology Data Corpus for Audio-Visual Speech Recognition. In *Euromedia'2009*, pp. 63–69. April 2009.

[Chi09b] A. G. Chiţu and L. J. M. Rothkrantz. Visual Speech Recognition - Automatic System for Lip Reading of Dutch. *Information Technologies and control*, vol. 7(3):pp. 2–9, 2009.

[Cıs04] P. Cısar, M. Zěleznẙ, and Z. Krnoul. 3D lip-tracking for audio-visual speech recognition in real applications. In *Prodeedings of the Interspeech*, pp. 2521–2524. 2004.

[Coi96] T. Coianiz, L. Torresani, and B. Caprile. 2d deformable models for visual speech analysis. In D. G. Stork and M. E. Hennecke, editors, *Speechreading by humans and machines : models, systems, and applications.*, vol. 150 of *NATO ASI Series F: Computer and Systems Sciences.* Springer, Berlin and New York, 1996.

[Coo92] T. Cootes and C. Taylor. Active shape models–smart snakes. In *Proc. British Machine Vision Conference*, pp. 266–275. Citeseer, 1992.

[Coo98] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *Proc. European Conference on Computer Vision 1998*, vol. 2, pp. 484–498. Springer, 1998.

[Coo00] T. Cootes. *Image Processing and Analysis*, chap. 7 Model-Based Methods in Analysis of Biomedical Images, pp. 223–248. Oxford University Press, 2000.

[Coo01] T. Cootes and C. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, vol. 4322, pp. 236–248. Citeseer, 2001.

[Cor84] P. Corthals. Een eenvoudige visementaxonomie voor spraakafzien [a simple viseme taxonomy for lipreading]. In *Tijdscrijf Log en Audio*, vol. 14, pp. 126–134. 1984.

[Cor02] M. Correia and A. Campilho. Real-time implementation of an optical flow algorithm. *Pattern Recognition*, vol. 4:p. 40247, 2002.

[D-C04] D-CIS LAB. Interactive Collaborative Information Systems, http://www.icis.decis.nl/, 2004.

[Dam94] M. Damhuis, T. Boogaart, C. Veld, M. Versteijlen, W. Schelvis, L. Bos, and L. Boves. Creation and analysis of the dutch polyphone corpus. In *Third International Conference on Spoken Language Processing*. ISCA, 1994.

[Dau03] P. Daubias and P. Deleglise. The lium-avs database: a corpus to test lip segmentation and speechreading systems in natural conditions. In *Eighth European Conference on Speech Communication and Technology*. Citeseer, 2003.

[Día06] J. Díaz, E. Ros, F. Pelayo, E. Ortigosa, and S. Mota. Fpga-based real-time optical-flow system. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16(2):pp. 274–279, 2006.

[Dry98] I. L. Dryden and K. V. Mardia. *Statistical shape analysis*. Wiley New York, 1998.

[Duc94] P. Duchnowski, U. Meier, and A. Waibel. See me, hear me: Integrating automatic speech recognition and lip-reading. *Reading*, vol. 1(1):pp. 1–2, 1994.

[Duc95] P. Duchnowski, M. Hunke, D. Büsching, U. Meier, and A. Waibel. Toward movement-invariant automatic lip-reading and speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing, 1995 (ICASSP-95)*, vol. 1, pp. 109–112. 1995.

[Dup00] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. In *IEEE Transactions On Multimedia*, vol. 2. September 2000.

[Edw98] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. In *3rd International Conference on Automatic Face and Gesture Recognition*, pp. 300–305. 1998.

[Egg64] J. P. M. Eggermont. *Taalverwerving bij een Groep Dove Kinderen [Language Acquisition in a Group of Deaf Children]*. 1964.

[Eve01] N. Eveno, A. Caplier, and P.-Y. Coulon. A new color transformation for lips segmentation. In *Multimedia Signal Processing, IEEE 4th Workshop on*, pp. 3–8. 2001.

[Eve04] N. Eveno, A. Caplier, and P.-Y. Coulon. Automatic and accurate lip tracking. In *IEEE Transactions on Circuits and Systems for Video technology*, vol. 15, pp. 706–715. May 2004.

[Fis68] C. G. Fisher. Confusions among visually perceived consonants. *Journal of Speech, Language and Hearing Research*, vol. 11(4):p. 796, 1968.

[Fit07] S. Fitrianie, R. Poppe, T. Bui, A. Chiţu, D. Datcu, R. Dor, D. Hofs, P. Wiggers, D. Willems, M. Poel, L. Rothkrantz, L. Vuurpijl, and J. Zwiers. A multimodal human-computer interaction framework for research into crisis management. *Proc of the Intelligent Human Computer Systems for Crisis Response and Management*, vol. 1:pp. 149–158, 2007.

[Fit10] S. Fitrianie, Z. Yang, D. Datcu, A. G. Chiţu, and L. J. M. Rothkrantz. *Interactive Collaborative Information systems*, vol. Studies in Computational Intelligence, chap. Context-Aware Multimodal Human-Computer Interaction, pp. 237–272. Springer, May 2010. ISBN 978-3-642-11687-2.

[Fle90] D. J. Fleet and A. D. Jepson. Computation of Component Image Velocity from Local Phase Information. *International Journal of Computer Vision*, vol. 5(1):pp. 77–104, August 1990.

[Fle00] D. J. Fleet, M. J. Black, Y. Yacoob, and A. D. Jepson. Design and Use of Linear Models for Image Motion Analysis. *International Journal of Computer Vision*, vol. 36(3):pp. 171–193, 2000.

[For73] G. D. Forney. The viterbi algorithm. *proc IEEE*, vol. 61(3):pp. 268–278, 1973.

[Fur03] S. Furui. Robust Methods in Automatic Speech Recognition and Understanding. In *EUROSPEECH 2003 - GENEVA*. 2003.

[G0̈0a] R. Göecke, J. B. Millar, A. Zelinsky, and J. Robert-Ribes. Automatic extraction of lip feature points. In *Proc. of the Australian Conference on Robotics and Automation ACRA2000*, pp. 31–36. Citeseer, 2000.

[G0̈0b] R. Göecke, Q. N. Tran, J. B. Millar, A. Zelinsky, and J. Robert-Ribes. Validation of an automatic lip-tracking algorithm and design of a database for audio-video speech processing. In *Proc. 8th Australian Int. Conf. on Speech Science and Technology SST2000*, pp. 92–97. Citeseer, 2000.

[G0̈4] R. Göecke and J. Millar. The audio-video australian english speech data corpus avozes. In *Proceedings of the 8th International Conference on Spoken Language Processing ICSLP2004*, vol. III, pp. 2525–2528. Jeju, Korea, October 2004.

[Gal98] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. In *Proceedings of the British Machine Vision Converence (BMVC) '98*. September 1998.

[Gan02] A. Ganapathiraju. *Support vector machines for speech recognition.* Ph.D. thesis, Mississippi State University, Mississippi State, MS, USA, 2002. Major Professor-Picone, Joseph.

[Gar88] J. Garofolo et al. Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database, 1988.

[Goo91] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society Series B (Methodological)*, pp. 285–339, 1991.

[Gra97] M. S. Gray, J. R. Movellan, and T. J. Sejnowski. Dynamic features for visual speechreading: A systematic comparison. *Advances in Neural Information Processing Systems*, vol. 9:pp. 751–757, 1997.

[Hee87] D. J. Heeger. Model for the extraction of image flow. *Journal Opt Soc Amer*, vol. 4(8):pp. 1455–1471, August 1987.

[Hil09] S. Hilder, R. Harvey, and B. J. Theobald. Comparison of human and machine-based lip-reading. In B. J. Theobald and R. W. Harvey, editors, *AVSP 2009*, pp. 86–89. Norwich, September 2009.

[Hon06] X. Hong, H. Yao, Y. Wan, and R. Chen. A pca based visual dct feature extraction method for lip-reading. *iih-msp*, vol. 0:pp. 321–326, 2006.

[Hor81] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, vol. 17:pp. 185–203, 1981.

[Hul98] A. Hulbert and T. Poggio. Synthesizing a color algorithm from examples. *Science*, vol. 239:pp. 482–485, 1998.

[Hun89] M. Hunt. Figures of merit for assessing connected-word recognisers. In *Speech Input/Output Assessment and Speech Databases.* ISCA, 1989.

[Iwa01] K. Iwano, S. Tamura, and S. Furui. Bimodal Speech Recognition Using Lip Movement Measured By Optical-Flow analysis. In *HSC2001*. 2001.

[Kas88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, vol. 1(4):pp. 321–331, 1988.

[Kir90] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, vol. 12(1):pp. 103–108, 1990.

[Kob92] H. Kobayashi and F. Hara. Recognition of mixed facial expressions by neural network. In *IEEE International Workshop on Robot and Human Communication, 1992. Proceedings.*, pp. 387–391. 1992.

[Kri08] R. Kricke, T. Gernoth, and R.-R. Grigat. Local binary patterns for lip motion analysis. In *Image Processing 2008, 15th IEEE International Conference on*, pp. 1472–1475. 2008.

[Kum07] K. Kumar, T. Chen, and R. M. Stern. Profile view lip reading. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing–ICASSP*, vol. 4, pp. 429–432. Citeseer, 2007.

[Lee04] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. Huang. Avicar: Audio-visual speech corpus in a car environment. In *INTERSPEECH2004-ICSLP*. Jeju Island, Korea, October 2004.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics-Doklady*, vol. 10. 1966.

[Li95] N. Li, S. Dettmer, and M. Shah. Lipreading using eigen sequences. In *Proc. International Workshop on Automatic Face- and Gesture-Recognition*, pp. 30–34. Zurich, Switzerland, 1995.

[Li97] N. Li, S. Dettmer, and M. Shah. Visually recognizing speech using eigensequences. *Motion-based recognition*, vol. 1:pp. 345–371, 1997.

[Lie99] M. Lievin, P. Delmas, P. Y. Coulon, F. Luthon, and V. Fristot. Automatic lip tracking: Bayesian segmentation and active contours in a cooperative scheme. In *IEEE Conference on Multimedia, Computing and Systems, ICMCS99*, vol. 1, p. 691696. Fiorenza, Italy, June 1999.

[Low76] B. T. Lowerre. The harpy speech recognition system. Tech. rep., Carnegie Mellon University, 1976.

[Luc81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, p. 674679. 1981.

[Luc06] P. Lucey and G. Potamianos. Lipreading using profile versus frontal views. In *IEEE Multimedia Signal Processing Workshop*, pp. 24–28. Citeseer, 2006.

[Lue96] J. Luettin, N. A. Thacker, and S. W. Beet. Statistical lip modelling for visual speech recognition. In *Proceedings of the 8th European Signal Processing Conference (EUSIPCO96)*. 1996.

[Lue97] J. Luettin and N. A. Thacker. Speechreading using probabilistic models. *Computer Vision and Image Understanding*, vol. 65(2):pp. 163–178, 1997.

[Mar95] A. Martin. Lipreading by optical flow correlation. Tech. rep., Compute Science Department University of Central Florida, 1995.

[Mar05] J. L. Martín, A. Zuloaga, C. Cuadrado, J. Lázaro, and U. Bidarte. Hardware implementation of optical flow constraint equation using fpgas. *Computer Vision and Image Understanding*, vol. 98(3):pp. 462–490, 2005.

[Mas91] K. Mase and A. Pentland. Automatic lipreading by optical-flow analysis. In *Systems and Computers in Japan*, vol. 22, pp. 67–76. 1991.

[Mat96] I. A. Matthews, J. Bangham, and S. J. Cox. Audiovisual speech recognition using multiscale nonlinear image decomposition. In *Fourth International Conference on Spoken Language Processing.* Citeseer, 1996.

[Mat02] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 198–213. 2002.

[McC05] I. McCowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, and H. Bourlard. On the use of information retrieval measures for speech recognition evaluation. *IDIAP (Institut Dalle Molle dIntelligence Artificielle Perceptive), IDIAP Research Report IDIAP-RR*, vol. 04:pp. 1–13, 2005.

[Mcg76] H. Mcgurk and J. Macdonald. Hearing lips and seeing voices. *Nature*, vol. 264:pp. 746 – 748, December 1976.

[Mes98] K. Messer, J. Matas, and J. Kittler. Acquisition of a large database for biometric identity verification. In J. Jan, J. Kozumplík, and Z. Szabó, editors, *BIOSIGNAL 98*, pp. 70–72. Vutium Press, Technical University Brno, Purkynova 188, 612 00, Brno, Czech Republic, June 1998.

[Mes99] K. Messer, J. Matas, J. Kittler, J. Lüttin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. In *Audio- and Video-based Biometric Person Authentication, AVBPA'99*, pp. 72–77. 1999. Washington, D.C., March 1999. 16 IDIAP–RR 99-02.

[Moo77] R. Moore. Evaluating speech recognizers. *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 25(2):pp. 178–183, 1977.

[Mor04] A. Morris, V. Maier, and P. Green. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Eighth International Conference on Spoken Language Processing.* 2004.

[Mor07] L. E. L. Morn and R. Pinto-Elas. Lips shape extraction via active shape model and local binary pattern. *MICAI 2007: Advances in Artificial Intelligence*, vol. 4827:pp. 779–788, 2007.

[Mov95] J. R. Movellan. Visual Speech Recognition with Stochastic Networks. In *Advances in Neural Information Processing Systems*, vol. 7. MIT Pess, Cambridge, 1995.

[Nag87] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, vol. 33(3):pp. 298–324, 1987.

[Nef02] A. V. Nefian, L. Liang, X. Pi, X. Pi, X. Liu, and K. Murphy. Dynamic bayesian networks for audio-visual speech recognition. *EURASIP Journal on Applied Signal Processing*, vol. 11:p. 12741288, 2002.

[Net00] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. Audio-visual speech recognition. In *Final Workshop 2000 Report*, vol. 764. Citeseer, 2000.

[Oja97] T. Ojala and M. Pietikainen. Unsupervised texture segmentation using feature distributions. *Image Analysis and Processing*, vol. 1310:pp. 311–318, 1997.

[Omo99] N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki. Time-compression: systems concerns, usage, and benefits. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, p. 143. ACM, 1999.

[Ozg08] E. Ozgur, B. Yilmaz, H. Karabalkan, H. Erdogan, and M. Unel. Lip segmentation using adaptive color space training. In *International Conference on Auditory and Visual Speech Processing*. Citeseer, 2008.

[Pat02] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy. CUAVE: A New Audio-Visual Database for Multimodal Human-Computer Interface Research. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2002.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, vol. 2(11):pp. 559–572, 1901.

[Pet88] E. Petajan, B. Bischoff, and D. Bodoff. An improved automatic lipreading system to enhance speech recognition. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 19–25. ACM Press, New York, NY, USA, 1988.

[Pig97] S. Pigeon and L. Vandendorpe. The M2VTS multimodal face database (release 1.00). *Lecture Notes in Computer Science*, vol. 1206:pp. 403–410, 1997.

[Pot97] G. Potamianos, E. Cosatto, H. Graf, and D. Roe. Speaker independent audio-visual database for bimodal ASR. In *Proc. Europ. Tut. Work. Audio-Visual Speech Proc., Rhodes*. 1997.

[Pot98] G. Potamianos, H. P. Graf, and E. Cosatto. An image transform approach for hmm based automatic lipreading. In *Proc. IEEE International Conference on Image Processing*, vol. 1, p. 173. Citeseer, 1998.

[Pot04] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 2004.

[Poy99] C. Poynton. Frequently asked questions about color. *http://wwwpoyntoncom/PDFs/ColorFAQpdf*, vol. -:pp. 1–24, 1999.

[Pr05] J. F. G. Prez, A. F. Frangi, E. L. Solano, and K. Lukas. Lip reading for robust speech recognition on embedded devices. In *Int. Conf. Acoustics, Speech and Signal Processing*, vol. I, p. 473476. 2005.

[Rab89] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77(2):p. 257, 1989.

[Sal07] A. Salazar, J. Hernandez, and F. Prieto. Automatic quantitative mouth shape analysis. *Lecture Notes in Computer Science*, vol. 4673:pp. 416–421, 2007.

[Sch10] T. Schultz. Weak and silent speech: Technologies to support people with speech impairment. In *12th International Conference on Computers Helping People with Special Needs*. Vienna University of Technology, Austria, July 12-13 2010.

[Sin91] A. Singh. Optic Flow Computation. A Unified Perspective. IEEE Computer Society Press, 1991.

[Sir87] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, vol. 4(3):pp. 519–524, 1987.

[Son94] N. van Son, T. M. I. Huiskamp, A. J. Bosman, and G. F. Smoorenburg. Viseme classifications of dutch consonants and vowels. *The Journal of the Acoustical Society of America*, vol. 96:p. 1341, 1994.

[Ste03] R. Sternberg. *Cognitive Psychology Third Edition*. 2003.

[Sun09] X. Sun, L. Rothkrantz, D. Datcu, and P. Wiggers. A bayesian approach to recognise facial expressions using vector flows. In *CompSysTech '09: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, pp. 1–6. ACM, New York, NY, USA, 2009.

[syn01] The SYNFACE project,
http://www.phon.ucl.ac.uk/home/andyf/synfaceUCL.htm, 2001.

[Tam02] S. Tamura, K. Iwano, and S. Furui. A robust multi-modal speech recognition method using optical-flow analysis. In *Extended summary of IDS02*, pp. 2–4. Kloster Irsee, Germany, June 2002.

[Tam04] S. Tamura, K. Iwano, and S. Furui. Multi-modal speech recognition using optical-flow analysis for lip images. *J VLSI Signal Process Syst*, vol. 36(2-3):pp. 117–124, 2004.

[Ter99] J.-C. Terrillon and S. Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. In *Vision Interface '99*. May 1999.

[Tom96] M. J. Tomlinson, M. J. Russell, and N. M. Brooke. Integrating audio and visual information to provide highly robust speech recognition. In *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 2. 1996.

[Tur91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, vol. 3(1):pp. 71–86, 1991.

[Ura88] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. In *Biological Cybernetics*, vol. 60, pp. 79–87. December 1988.

[Var93] A. Varga and H. J. M. Steeneken. Assessment for automatic speech recognition II: NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Commun*, vol. 12(3):pp. 247–251, 1993.

[Vio01] P. Viola and M. Jones. Robust Real-time Object Detection. In *Second International Workshop On Statistical And Computational Theories Of Vision Modeling, Learning, Computing, And Sampling*. Vancouver, Canada, July 2001.

[Vis99] M. Visser, M. Poel, and A. Nijholt. Classifying visemes for automatic lipreading. *Lecture notes in computer science*, vol. 0:pp. 349–352, 1999.

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, vol. 13(2):pp. 260–269, 1967.

[Wan07] S. L. Wang, W. H. Lau, A. W. C. Liew, and S. H. Leung. Robust lip region segmentation for lip images with complex background. *Pattern Recognition*, vol. 40(12):pp. 3481–3491, 2007.

[Wax88] A. Waxman, J. Wu, and F. Bergholm. Convected activation profiles and receptive fields for real time measurement of short range visual motion. In *Proceedings of Conference Computational Visual Pattern Recognition*, pp. 771–723. 1988.

[Wig02] P. Wiggers, J. Wojdel, and L. Rothkrantz. Medium vocabulary continuous audio-visual speech recognition. In *ICSLP, Conference Proceedings of*. September 2002.

[Wig08] P. Wiggers. *Modelling context in automatic speech recognition*. Ph.D. thesis, Delft University of Technology, 2008.

[Wil97] J. J. Williams, J. C. Rutledge, D. C. Garsteckiy, and A. K. Katsaggelos. Frame rate and viseme analysis for multimedia applications. In *Proc. IEEE Works. Multimedia Signal Process., Princeton*, pp. 13–18. 1997.

[Wil98a] J. J. Williams, J. C. Rutledge, and A. K. Katsaggelos. Frame rate and viseme analysis for multimedia applications to assist speechreading. *Journal of VLSI Signal Processing*, vol. 20:pp. 7–23, 1998.

[Wil98b] J. R. Williams. Guidelines for the use of multimedia in instruction. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 42, pp. 1447–1451. Human Factors and Ergonomics Society, 1998.

[Woj00] J. C. Wojdel and L. J. M. Rothkrantz. Visually based speech onset/offset detection. In *Proceedings of 5th Annual Scientific Conference on Web Technology, New Media, Communications and Telematics Theory, Methods, Tools and Application (Euromedia 2000)*, pp. 156–160. Antwerp, Belgium, 2000.

[Woj02] J. Wojdel, P. Wiggers, and L. Rothkrantz. An audio-visual corpus for multimodal speech recognition in dutch language. In *ICSLP, Conference Proceedings of.* 2002.

[Woj03] J. C. Wojdel. *Automatic Lipreading in the Dutch Language.* Ph.D. thesis, Delft University of Technology, November 2003.

[Yos03] T. Yoshinaga, S. Tamura, K. Iwano, and S. Furui. Audio-Visual Speech Recognition Using Lip Movement Extracted from Side-Face Images. In *AVSP2003*, pp. 117–120. September 2003.

[Yos04] T. Yoshinaga, S. Tamura, K. iwano, and S. Furui. Audio-visual speech recognition using new lip features extracted from side-face images. In *Robust 2004.* August 2004.

[You05] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.4).* Citeseer, 2005.

[Zha00] X. Zhang and R. Mersereau. Lip feature extraction towards an automatic speechreading system. In *ICIP00.* Vancouver, Canada, September 2000.

[Zha02] X. Zhang, C. C. Broun, R. M. Mersereau, and M. A. Clements. Automatic speechreading with applications to human-computer interfaces. *EURASIP J Appl Signal Process*, vol. 2002(1):pp. 1228–1247, 2002.

[Zha07] G. Zhao, M. Pietikäinen, and A. Hadid. Local spatiotemporal descriptors for visual recognition of spoken phrases. In *Proceedings of the international workshop on Human-centered multimedia*, pp. 66–75. ACM, 2007.

Summary

"Towards Robust Visual Speech Recognition"

In the last two decades we witnessed a rapid increase of the computational power governed by Moore's Law. As a side effect, the affordability of cheaper and faster CPUs increased as well. Therefore, many new "smart" devices flooded the market and made informational systems widely spread. The number of users of information systems has also increased many folds, and the user's characteristics have changed to include not only a small number of initiates but also a majority of non technical people. To make this transition possible systems' developers had to change the computer user interfaces in order to make it simpler and more intuitive. However, the interaction was still based on rather artificial devices such as mouse and keyboard. Since the Moore's Law continues to work over and over again we came to a critical moment when the current systems can easily cope with other input streams such as video and audio, to name the most important, and others. We can, therefore, envision systems with which we can communicate through speech and body movements and that can automatically and transparently adapt to the environment and user. This can be done for instance by recognizing the user affective state, by understanding the task of the user and recognizing the context of the interaction.

Automatic speech recognition by capturing and processing the audio signal is one development in this direction. Even though in controlled settings automatic speech recognition has achieved spectacular results, its performance is still dependent on the context (for instance on the level of the background noise). Automatic lip reading has appeared in this context as a way to enhance automatic speech recognition in noisy environments. Even though it is still a relatively novel research domain, other applications were found which employ lip reading as stand alone: interfaces for hearing impaired persons, security applications, speech recovery from mute of deteriorated films, silence interfaces.

With the advances in visual signal processing the research in lip reading was also revitalized. However, at the moment of writing of this thesis lip reading was still waiting for its great leap. This thesis investigates several techniques for directing lip reading towards more robust performances.

The thesis starts by introducing the relevant methodologies that govern automatic lip reading. Next it introduces all the concepts needed to understand the technologies, experiments, results and discussions presented later on. It is, therefore, one of the most important parts of the thesis. The presentation of the state of the art in lip reading is setting the starting point of the research presented. Before, continuing to follow the lip reading process the thesis introduces the mathematical foundations that give the theoretical support for the analysis.

All our systems are based on the Hidden Markov Models approach. This paradigm has proved to be very useful in similar problems and we successfully employed it for lip reading. The main idea behind it is the Bayesian rule which says that starting from some a-priori knowledge we can always improve our understanding of a system through observation. Observation translates into processing the video stream in a set of features that describe what is being said by the speaker. However, in order to appropriately train lip reading systems, a large amount of data is needed. The first important contribution of our research is a large data corpus for the Dutch language. This corpus, named "New Delft University of Technology Audio Visual Speech Corpus", is at the date of writing this thesis one of the largest corpora for lip reading in Dutch. The corpus contains dual view high speed recordings (i.e. 100Hz) of continuous speech in Dutch. During the building of the corpus, we also produced an incipient set of guidelines for building a data corpus for lip reading which we hope to be useful for other researchers.

However, the core of this thesis consists in the data parametrization. Data parametrization is the process that transforms the input video data in a set of features that are used later on for training and testing the resulting recognizer. The parametrization should reduce the size of the input data while preserving the most important information related with what the speaker says. We investigated three data parametrization techniques each coming from a different category of algorithms. We used Active Appearance Models which generate a combined geometric and appearance based set of features, we used optical flow analysis which is an appearance based approach that directly accounts for the apparent movement on the speaker's face and we used a statistical approach which generates the geometry of lips without starting from an a-priori fixed model. During the research presented in this thesis we investigated the performances of these data parametrization techniques and we pointed out their strengths and weaknesses. We also analysed the performance of lip reading starting from other points of view. We analysed the influence of the sampling rate of the video data, the performance of the lip readers as a function of the recognition task but also the performance as a function of the size of the corpus used. Answering to all these questions improved our understanding of the limitations and the possible improvements of lip reading.

Alin G. Chiţu

Samenvatting

“Op weg naar robuuste visuele spraakherkenning”

In de laatste twee decennia zagen we een snelle toename van de rekencapaciteit volgens de wet van Moore. Als gevolg hiervan kwamen goedkope en snelle CPU’s beschikbaar. Daardoor kwamen vele nieuwe “slimme” apparaten op de markt en raakten informatiesystemen wijdverbreid. Het aantal gebruikers van informatiesystemen is vele malen toegenomen waarbij de gebruikers zijn veranderd van specialisten tot een grote hoeveelheid niet-technische geschoolde mensen. Om deze overgang mogelijk te maken moesten systeemontwikkelaars de gebruikersinterface veranderen om de interactie simpeler en meer intuïtief te laten verlopen. Echter de interactie was nog steeds gebaseerd op nogal onnatuurlijke randapparatuur zoals een muis en een toetsenbord. Op dit moment is de rekencapaciteit van gangbare systemen dusdanig dat ze gemakkelijk kunnen omgaan met andere datastromen zoals beeld en geluid als meest belangrijke. In de nabije toekomst kunnen we systemen verwachten waar we mee kunnen communiceren via spraak en lichaamsbewegingen en die zich automatisch en transparant kunnen aanpassen aan de omgeving van de gebruiker. Dit kan bijvoorbeeld plaatsvinden door herkenning van de emotionele toestand van de gebruiker, door herkenning van de taken en herkenning van de context waarbinnen de interactie plaatsvindt. Automatische spraakherkenning door opname en verwerking van het geluidsignaal is een van de ontwikkelingen in deze richting. Alhoewel automatische spraakherkenning in gecontroleerde omgevingen spectaculaire resultaten heeft bereikt, is het resultaat nog steeds sterk afhankelijk van de context (bijvoorbeeld het niveau van achtergrondruis). Automatische liplezen biedt de mogelijkheid om spraakherkenning te verbeteren in omgevingen met veel achtergrondruis. Andere toepassingsgebieden, waarbij spraakherkenning als apart systeem wordt gebruikt, zijn: interfaces voor auditief gehandicapte personen, applicaties in beveiligde omgevingen, spraak reconstrueren vanuit stomme of beschadigde films, of geluidloze interfaces. Door de vooruitgang in visuele signaalverwerking is het onderzoek binnen liplezen ook gerevitaliseerd. Echter op het moment van het schrijven van deze thesis, was liplezen in afwachting van de grote sprong vooruit. In dit proefschrift worden verschillende technieken onderzocht waardoor liplezen meer robuust wordt.

Dit proefschrift start met de introductie van relevante methodologieën die toegepast worden binnen liplezen. Vervolgens worden alle begrippen geïntroduceerd die nodig zijn om de technologieën, experimenten, resultaten en discussies op het eind van dit proefschrift te begrijpen. Het is daarom een van de belangrijkste onderdelen van dit proefschrift. De presentatie van de huidige stand van zaken binnen liplezen is de start van het onderzoek. Eerst wordt de wiskundige basis geïntroduceerd die de theoretische onderbouwing verschaft van de analyse.

Al onze systemen zijn gebaseerd op hidden Markov model benadering. Dit paradigma is zeer nuttig gebleken in vergelijkbare problemen en we hebben het met succes toegepast voor liplezen. De voornaamste achterliggende gedachte is de regel van Bayes, de zegt dat als gestart wordt vanuit a priori kennis we onze kennis van het systeem altijd kunnen verbeteren met behulp van observaties. Observatie wordt vertaald in het extraheren van kenmerken uit de video-opnames die beschrijven wat er gezegd is door spreker. Echter om een lipleessysteem te trainen zijn grote hoeveelheden data nodig. De eerste belangrijkste bijdrage van ons onderzoek is een groot databestand van de Nederlandse taal. Dit databestand wordt “New Delft University of Technology Audio Visual Speech Corpus” genoemd, en is op het moment van het schrijven van dit proefschrift een van de grootste databestanden voor liplezen voor de Nederlandse taal. Het databestand bevat duale (frontale/silhouet gezichtsopnames) hoge snelheidsvideo-opnames (d.w.z. 100Hz) van continue spraak in het Nederlands. Tijdens het bouwen van het databestand hebben we ook een verzameling van richtlijnen opgesteld hoe een databestand voor liplezen opgebouwd moet worden waarvan we hopen dat het waardevol is voor andere onderzoekers. Echter de kern van het proefschrift wordt gevormd door de data parametrisering. Data parametrisatie is het proces dat video data transformeert naar een verzameling kenmerken die later gebruikt worden bij het trainen en testen van de uiteindelijke herkenner. De parametrisatie wordt geacht de omvang van de input data te reduceren, met behoud van de meest belangrijke informatie over hetgeen wat de spreker zegt. We onderzochten drie dataparametrisatie technieken die elk afkomstig zijn uit verschillende categorieën van algoritmen. We gebruikten Active Appearance modellen, die een set kenmerken genereren die een combinatie vormen van geometrische en visuele kenmerken, vervolgens gebruikten we optische technieken uit de stromingsleer die een visueel gebaseerde benadering is, die direct bijdraagt aan de visuele bewegingen op het gezicht van de spreker. Daarnaast gebruikten we een statistische benadering, die de geometrie van de lippen genereert zonder te starten van uit een a priori model. Tijdens het onderzoek gepresenteerd in dit proefschrift onderzochten we hoe goed de verschillende dataparametrisatie technieken waren en we bepaalden de sterke en zwakke punten. We analyseerden de prestaties van het lipleessyteem ook vanuit andere gezichtspunten. We analyseerden de invloed van de snelheid bij de verschillende video opnamen, de werking van de lipleessystemen als een functie van de herkenningstaak maar ook als functie van de omvang van het databestand. Bij het beantwoorden van al deze vragen kregen we een beter inzicht in de beperkingen en mogelijke verbeteringen van lipleessystemen.

Alin G. Chiţu

Acknowledgements

I am finally here! It was fascinating, challenging, difficult, overwhelming, tiring, sometimes I felt hopeless, but then again I felt I got new wings. Even though there are only four years (OH!, ok almost five) I feel I could write an entire book only about the life as a PhD candidate. The lesson I have learned the most is that while you're pursuing your doctoral degree your own life goes on as well and there is no way you could survive alone. You need some people to be close to you and help you during your tough moments, but also be happy with you during your peaks; you need to live your life! Therefore, the time has come to acknowledge and thank all the people that were close to me during these years of my life and in many ways contributed to my becoming the person I am today and contributed directly or indirectly to this thesis.

First of all, I would like to thank my promoter, Leon Rothkrantz, for giving me the opportunity to work on this fascinating subject and for his supervision during all these years. I thank him for his understanding and support during those periods that were so difficult for me and my family. I also thank him for the weekly discussions that kept me on the right track and for being on my side during the last months before coming to this happy ending.

I would also like to thank my promoter, Catholijn Jonker, for the time she invested for completing this thesis. Thanks to her valuable comments I was able to improve the thesis both in writing style but also in readability. Thank you again for your understanding during the last year of my endeavour.

This is a good moment to thank all the committee members for their valuable comments which improved this thesis in many ways. Thank you all.

None of the work presented in this thesis would have been possible without the data corpus compiled with so much work. I would like to thank all the participants to our long and tiring recording sessions. I would like to especially thank Ank Voets which was so kind to record no less than 18 sessions with us. Thank you very much! Your effort was so valuable to me and I am sure it will be for future researchers as well. I also want to thank Karin Driel, Pegah Takapoui and Mathijs van Vulpen for their valuable help during building the language pool, setting the recording environment and supervising the recordings.

Since this dissertation is such an important milestone of my (academic) career but also of my life in general and especially of my life here in The Netherlands, I think this is a good moment to thank everybody that helped us (me and my wife) from the first moment of our arrival here in The Netherlands.

Now, when I am writing this thesis I have a confusing feeling: I feel like there are ages ago since we first came here but in the same time I remember everything as it was yesterday. I remember exactly the day we took the bus and embarked on the 48 hours journey from Bucharest to Rotterdam. I remember even more vividly the strong feelings of excitement we had in our hearts. It feels like it was yesterday when we arrived in Rotterdam at three in the night and Remus was waiting for us. His optimism and thirst for life was always inspiring and, more importantly, has always improved our moral. I always felt better after visiting and talking with him and with his wife Dana. After every discussion with Remus, you always feel like life is easier, things are better or they go towards the right direction. I want to thank you both for being close to us and for all the good time we spent together.

It is impossible for me not mention here Roger Cooke. I would like to thank again (I did that as well with my master thesis) for trusting me and giving me the opportunity to enrol to the International Master Program at Delft University. This has changed my life completely. He was the first real scientist I met that was exactly as I imagined and as I admired since I was a little boy and watched Jacques-Yves Cousteau's films at "Teleenciclopedia" - the 45 year old tv documentary show broadcasted on the national Romanian TV channel. It was during this master program when I felt for the first time that I am so close to science. I want to thank again to all the other people that planted the science seed in me and for their influence in my development as a scientist: Arnold Heemink, Tom Mazzuchi, Jan van Noortwijk, Peter Wilders, Dorota Kurowicka, etc. just to name the most important ones.

I also would like to thank all my friends and colleagues from TUDelft for the relaxing time we spent together, for the talks, lunches, picnics, trips, "cola-quium" and parties we organised together and for helping me become the person I am today. They have touched my life in irreversible ways. They are many and trying to list them all would always leave some out. Therefore, not to offend any of them I want to thank them all and they will know who they are. I particularly want to thank Dragoş Datcu for being my friend and for the long talks we had during these years, scientific or not, at the swimming pool and sauna or otherwise. I would like to thank as well Siska Fitrianie. When I first came to the Man-Machine Interaction Group she was the person that made my integration in the group a lot smoother. She was the person to ask about everything I needed to know. I have always enjoyed talking with her and I have always felt that she understood me even though we came from such different backgrounds.

At this important milestone in my life, I also what to thank my good friend Jonathan-Jean Stern. Even though we saw each other so rarely in the last 7 years, my first meeting with him came in a particular important moment of my life and left such a strong impression on me that changed me completely. I thank you for the long talks we had and for teaching me that people need to have “a gram of humility” in order to be good people.

Finally, I want to thank my family.

I would first like to thank my parents for all their continuous effort in raising me and for making following my dreams possible. I want to thank them for supporting me all these years away from home. I now realize that I left home after graduating high school and since then came home only for short holidays. I want to especially thank my mother which in her heart was never content with the idea that I was so far away and that we visited each other so rarely. (Sărut-mâna dragii mei părinţi!)

I want to thank my brother for being a model for me ever since I was a little boy. I was so glad when he and my sister in law moved to Brussels and we end up living even closer from each other than we were in Romania.

I want to thank my wife Dana from all my heart. Without her nothing would have been possible. I thank her for being close to me at all times, during my ups and downs, which were so many, I know. She has always shown me the bright side of life and made me believe that things would be better and so gave me the strength to continue. In the last years we went through many difficult times but together we succeeded to surpass them. There were also wonderful moments which I was so fortunate to share with her. During this time our beautiful daughters were born. There were many days when she was alone with them because I had to go to the office and work even though it was late after office hours or weekend. I thank you for those days and I am in debt for that. For everything and for every moment we spent together, I want to thank you, Dana! (Te iubesc puişorul meu!) I want to thank our sweetest daughters Mina and Ana. Not being, yet, able to understand why is daddy not playing with them all the time and why does he always go into the small bedroom, which we called “home office”, and not stay with them in the living room to play, they suffered the most during these times. I want to ask their forgiveness for all the times I was too tired and to stressed and did not have enough energy to answer their questions and maybe even sent them away, nervously. I want to assure them that I will always believe that they are my highest achievement in this world, and that I would not trade the time I spent with them for anything in this world. They have complicated and enriched our lives in ways we never have thought of. (Tati va iubeşte din suflet!)

Alin G. Chiţu
Delft, August 2010

Curriculum Vitae

Alin Gavril Chiţu was born in Buşteni, Romania on November 8, 1978.

He finished the secondary school in 1993 and the high school in 1997 in his home town. Starting from October 1, 1997 he studied Mathematics and Computer Science at University of Bucharest. He graduated in 2001 and went on for his first master program on Applied Computer Science at the same university which he graduated in February 2003. From February to July 2002 and from March to July 2003 he was enrolled in an exchange program in the Graphics Lab at the Computer Science Faculty of the National University of Singapore (NUS). In November 2003 there was a great opportunity for him to join the International Master Program in Risk Management and Environmental Modelling at Delft University of Technology. He graduated with honours in August 2005 with the thesis "Probability of ship collision with offshore wind farms in the Southern North Sea". His thesis was mentioned in the first three best master theses at the "Risk Management Study Award 2006". Starting on September 1, 2005 he went on to become a PhD candidate (in Dutch: *Assistent in Opleiding* or simply AIO) at the Faculty of Electrical Engineering Mathematics and Computer Science at the Delft University of Technology as a member of the Man-Machine Interaction Group under the daily supervision of Prof. dr. drs. L.J.M. Rothkrantz.

Starting from February 2010 he is working as a scientist in the Oil and Gas department at TNO (Netherlands Organisation for Applied Scientific Research) in the Built Environment and Geosciences core area.

Alin is married and has two beautiful twin daughters.

Colophon

This manuscript was typeset by the author with the $\LaTeX\,2_{\varepsilon}$ Documentation System on a PC running Windows XP SP3 and Cygwin.

Text editing was done using WinEdt and compiled using the MikTeX toolkit. The graphs and the diagrams were created in MatLab and Microsoft Visio, respectively. The conversion to Encapsulated Post Script (.eps) was done via “Print to PDFCreator”. Some illustrations were created through direct screen copy using the PrntScrn function and processed using *Adobe Photoshop.* The cover was produced by the author using *Adobe Illustrator.*

The body type is 10 point Computer Modern Roman. Chapter and section titles are in various sizes of Adobe Helvetica-Narrow Bold. The mono-space typeface used for verbatim text is Adobe Courier.

The $\LaTeX\,2_{\varepsilon}$ page formatting is a modified version of the “It Took Me Years To Write” template by Leo Breebaart[1]. The author would like to extend his gratitude to Mr. Leo Breebaart for this valuable resource.

[1] Leo Breebaart, “It Took Me Years To Write” template for Delft University PhD Thesis, 2003. Downloadable from: http://www.kronto.org/thesis/ (the link was accessible in May 2010)